DOCUMENT RESUME

ED 088 205 EA 005 939

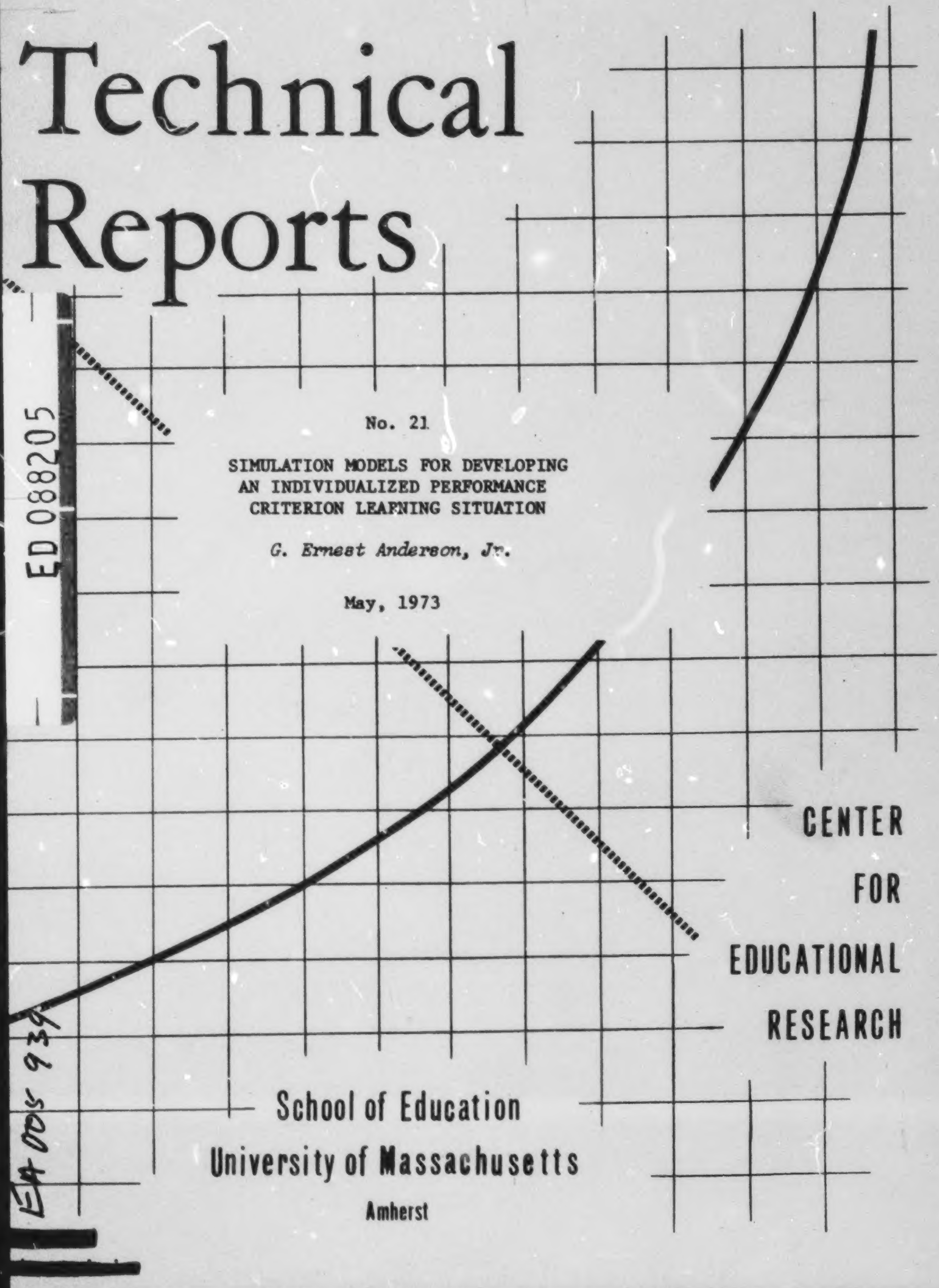
AUTHOR	Anderson, G. Ernest, Jr.
TITLE	Simulation Models for Developing an Individualized, Performance Criterion Learning Situation. Technical Monograph No. 21.
INSTITUTION	Massachusetts Univ., Amherst. School of Education.
SPONS AGENCY	General Learning Corp., Washington, D.C.; Office of Education (DHEW), Washington, D.C.
BUREAU NO	BR-0-A-023
PUB DATE	May 73
GRANT	OEG-0-9-310417-4040; OEG-1-70-0010(509)
NOTE	243p.; Documentation of Queue, Adult Resources Flow, Edsim 1, Edsim 2, and Edsim 4
EDRS PRICE	MF-$0.75 HC-$11.40
DESCRIPTORS	Computer Assisted Instruction; *Computer Oriented Programs; Computer Programs; Computers; *Educational Development; Elementary Schools; Facility Utilization Research; Individualized Instruction; *Individualized Programs; Models; *Performance Criteria; Research Projects; Resource Allocations; *Simulation; Staff Utilization; Teacher Education
IDENTIFIERS	Model Elementary Teacher Education Project

ABSTRACT

The mission of the simulation team of the Model Elementary Teacher Education Project, 1968-71, was to develop simulation tools and conduct appropriate studies of the anticipated operation of that project. The team focused on the experiences of individual students and on the resources necessary for these experiences to be reasonable. This report contains documentation of QUEUE, an elementary queueing model useful in exploring how many channels of a specific service facility should be available to meet a hypothesized set of student needs; ARF 1, a system used to study staffing patterns and work loads; EDSIM 1, a method of keeping track essentially of student time to completion and student time in each instructional area; EDSIM 2, the major model for the Model Elementary Teacher Education proposed program, which keeps track of individual student progress as well as resources needed under various circumstances; and EDSIM 4, a model addressing the same problems as EDSIM 2 but from a different approach. (Computer printouts and charts may reproduce poorly.) (Author)

BEST COPY AVAILABLE

Technical Reports



No. 21

SIMULATION MODELS FOR DEVELOPING AN INDIVIDUALIZED PERFORMANCE CRITERION LEARNING SITUATION

G. Ernest Anderson, Jr.

May, 1973

CENTER FOR EDUCATIONAL RESEARCH

School of Education
University of Massachusetts
Amherst

ED 088205

EA 005 939

Technical Monograph No. 21

Project No. 0-A-023
Grant No. OEG-1-70-0010 (509)

SIMULATION MODELS FOR DEVELOPING AN INDIVIDUALIZED, PERFORMANCE CRITERION LEARNING SITUATION

Documentation of:
- Queue
- Adult Resources Flow
- Edsim 1
- Edsim 2
- Edsim 4

G. Ernest Anderson, Jr.

May, 1973

The research reported herein was performed pursuant to a grant with the Office of Education, U.S. Department of Health, Education, and Welfare. Contractors undertaking such projects under Government sponsorship are encouraged to express freely their professional judgement in the conduct of the project. Points of view or opinions stated do not, therefore, necessarily represent official Office of Education position or policy.

U.S. DEPARTMENT OF
HEALTH, EDUCATION, AND WELFARE

Office of Education
Bureau of Research

EA 005 939

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

TABLE OF CONTENTS

SECTION I - QUEUE..1-11
    - Description
    - Operating Instructions
    - Flow Chart
    - Program

SECTION II - ADULT RESOURCES FLOW..1-37
    - Description
    - Batch Version
    - Time Sharing Version

SECTION III - EDSIM I..1-20
    - Description
    - Operating Instructions
    - Program

SECTION IV - EDSIM II..1-88
    - Description
    - Flow Charts
    - Programs

SECTION V - EDSIM IV...1-50
    - Description
    - Program Documentation
    - Programs

Acknowledgements

The mission of the Simulation Team of the Model Elementary Teacher Education Project, 1968-1971, was to develop simulation tools and conduct appropriate studies of the anticipated operation of the Model Elementary Teacher Education Project.[1] The team's focus was on the experiences of individual students, and the resources necessary for these experiences to be reasonable.

Because of many operational similarities, it was possible to assist in the development of the first educational facility for the Fort Lincoln New Town Project in Washington, D.C., with the support of General Learning Corporation. Some of the model development work reported here resulted from that context.

This report contains several sections, as follows:

1. Documentation of QUEUE, an elementary queueing model useful in exploring the number of service channels needed to meet a set of random demands and random service times.
2. Documentation of ARF 1, used to study staffing patterns and work loads.

---

[1] Creation of these models was supported by the Model Elementary Teacher Education Project at the University of Massachusetts, under the direction of Dr. James M. Cooper, primarily as part of the Phase II Feasibility Study performed under U.S.O.E. contract OEG-0-9-310417-4040 (010).

Minor additional support was obtained from General Learning Corporation in conjunction with planning for the Fort Lincoln New Town educational facilities in Washington, D.C.

EDSIM 1 and 2 were created primarily by Mr. George F. Williams and Mr. Wayne E. Leininger, under the guidance of Dr. Eugene E. Kaczka, all of the School of Business Administration, University of Massachusetts.

EDSIM 4 was created by Mr. Thomas C. Richards, as his Ph.D. thesis, under the guidance of Dr. Edward J. Rising of the Industrial Engineering Department, University of Massachusetts.

These models are separately described in various documents.

3. Documentation of EDSIM 1, which keeps track essentially of student time to completion and student time in each instructional (pedagogical) area.

4. Documentation of EDSIM 2, which is the major model for the Model Elementary Teacher Education proposed program, keeping track of individual student progress as well as resources needed under various circumstances.

5. Documentation of EDSIM 4, which is a model addressing the same problems as EDSIM 2 but from a different approach.

The work reported here would not have been possible without the support of many people. Dr. James M. Cooper and Mr. Milton H. Ojala, Director and Assistant Director respectively of the METEP Project, have provided continuing support and encouragement. Dr. Joseph I. Lipson and Dr. Katherine O'Keefe, consultants to the Fort Lincoln New Town Project of General Learning Corporation, provided incentive and data for model development in that context, and helped set the stage for the model validation work at the Learning Research and Development Center, University of Pittsburgh. Dr. Richard Ferguson and the staff of the Learning Research and Development Center made data available and provided valuable feedback in regard to the value and use of simulation models. Miss Katherine Cowles and the staff of the Research Computing Center were of great assistance in seeing the various computer models through both machine and programming errors.

The members of the Simulation Team have contributed much to each other's work. Nonetheless, it is appropriate to recognize each individual's contribution:

Dr. G. Ernest Anderson, Jr., Associate Professor of Education. Team Coordinator. Primary responsibility for the ARF Model.

Dr. Eugene E. Kaczka, Associate Professor of Business Administration. Primary responsibility for data interpretation, and guidance in the creation of EDSIM 1 and 2.

Dr. Edward J. Rising, Professor of Industrial Engineering. Primary responsibility for guidance in the creation of EDSIM 4, and for interpreting simulation to many others in the METEP program. Also provided considerable assistance in other phases of the feasibility study.

Mr. Wayne Leininger, doctoral student in Business Administration. Co-creator of EDSIM 1 and 2. Also participated in the cost analysis studies of METEP.

Mr. Thomas Richards, doctoral student in Industrial Engineering. Creator of the QUEUE program and EDSIM 4. Major responsibility for improving data collection techniques and data editing.

Mr. George Williams, doctoral student in Business Administration. Co-creator of EDSIM 1 and 2.

Mr. William Foley, doctoral student in Education. Performed validation work of the EDSIM 1 model.

Mr. Frederick deFriesse, undergraduate student in Industrial Engineering. Valuable assistance where needed, including data editing, programming, and debugging.

QUEUE

This program samples a given arrival rate of students and a given service rate for these students, both from a Poisson distribution, and calculates student waiting time and facility usage for every level of facility availability between the level specified and 1. The intended use of this program is to explore how many of a specific service facility should be available to meet a hypothesized student need. A typical problem being investigated may be described as follows:

> Given an individualized curriculum that will cause certain relatively random demands on various facilities (as against more typical carefully scheduled demand, in which case a student has a fixed amount of time to do whatever he can), how many of each kind of facility should be available? (Guidance Counselors, Arts & Craft Stations, etc.)

The answer to such problems really involve administrative trade-off decisions between investment in facilities and the freedom implied in individualized curricula (as measured by student waiting time). Stated another way, there is a compromise to be made between use of facilities and use of student time. The QUEUE program does not by itself indicate what compromise to make; it merely indicates the probable effect of having a specified number of facilities available to meet a hypothesized demand.

On the following pages are the results of a sample problem, simulating a 9 hour day, with 20 students per hour arriving on the average, each requiring about 10 minutes of service time, individually (group size 1). Initially, 20 service facilities were hypothesized.

Each line of the output represents a one-day sample. Shown is the number of service facilities available that day, the average student waiting time (assuming all arriving students wait until they are serviced or the day ends), the average (per cent) facility utilization, and the number of students serviced.

The first sample run started with a hypothesized 20 service facilities. It shows that the needs could be met with nine or fewer facilities; consequently, three more sample runs were made starting with this number. Although the output differs somewhat from run to run, as is expected in a simulation exercise, the emerging pattern is that most, if not all, students can be accommodated with four heavily used service facilities. However, if student waiting time is important, 5 or 6 facilities should be considered, even though these facilities will then be used a smaller per cent of the time.

Interpretation of the results of QUEUE depends on several factors not considered in the computer program. To have a given level of service facilities available to students at any given time may require actually having additional facilities, considering maintenance. The operation of a school may not permit really random student arrivals: certain times (e.g. lunch) may be set aside for other activities, or, curricular considerations may indicate an uneven demand for a service facility.

Service time may also require careful thought. There is relatively little experimental evidence about how long children will spend with a given device, or, for that matter, about how often they will select it, in an individualized environment that offers considerable freedom.

Operating Instructions: (UMass Time - Sharing Version)

1. FETCH the program, or read it in on punched tape.
2. Replace the test data in line 900 with the data to be used. Additional lines of data may also be keyed, with line numbers greater than 720 but less than 900. The program will operate on successive sets of data until the ending conditions of line 901 are encountered.

   The input data are:

   a. Arrival rate, students per hour, F format. It may be desirable to select a peak hour rather than a typical hour for some purposes. The model will select actual arrivals from a Poisson distribution with the given arrival rate as its parameter.
   b. Service rate, minutes per student, F format. The model will select actual service rates from a Poisson distribution with the given service rate as its parameter.
   c. Group size (how many students are to be serviced at once), F format.
   d. The starting number of service facilities, I format. This is the largest number of service facilities that might be needed. The model will give the consequences of having one through this number of facilities.
   e. Length of the day, in hours, F format.
   f. Number of days to be simulated, F format. Currently, this is limited to 1.0; different runs can give a picture of different sample days.

   Sample input data line:

   ```
   900 20.0, 10.0, 1.0, 9, 9.0, 1.0
   ```

3. Decide whether or not to use the random number generator seed contained in the program at line 25. Restarting the program with different data without changing this seed will mean that the random number generator will start at the same place for each run, and, consequently, any idiosyncracies of the generator will affect each run in exactly the same way. To change the seed for the random number generator, rekey line 25 with the new seed.

   Sample random number generator seed line:

   ```
   25 ISEED = 975384763
   ```

4. Give the RUN command.

```
ARRIVAL RATE,STUDENTS PER HOUR= 20.00
SERVICE RATE,MINUTES PER STUDENT= 10.00
GROUP SIZE=  1.00
STARTING NUMBER OF SERVICE FACILITIES= 20
LENGTH OF DAY IN HOURS=  9.00
NUMBER OF DAYS IN SIMULATION=  1.00


NUMBER OF   AVERAGE  AVERAGE  TOTAL
SERVICE     STUDENT  FACILITY NUMBER
FACILITIES  WAITING  USEAGE   STUDENTS
            TIME              SERVICED
   20            0    16.37   178.00
   19            0    18.88   185.00
   18            0    18.20   164.00
   17            0    20.68   168.00
   16            0    20.97   165.00
   15            0    22.68   181.00
   14            0    24.42   171.00
   13            0    27.31   181.00
   12            0    24.86   158.00
   11            0    24.85   165.00
   10            0    29.25   175.00
    9          .00    40.86   186.00
    8          .02    40.72   163.00
    7          .32    51.34   172.00
    6         2.18    59.99   188.00
    5         1.52    65.80   179.00
    4         5.95    82.82   181.00
    3        53.12    97.05   129.00
    2       108.46    99.43   106.00
    1       188.01    99.84    57.00
```

```
ARRIVAL RATE,STUDENTS PER HOUR= 20.00
SERVICE RATE,MINUTES PER STUDENT= 10.00
GROUP SIZE=  1.00
STARTING NUMBER OF SERVICE FACILITIES=  9
LENGTH OF DAY IN HOURS=  9.00
NUMBER OF DAYS IN SIMULATION=  1.00


NUMBER OF   AVERAGE   AVERAGE   TOTAL
SERVICE     STUDENT   FACILITY  NUMBER
FACILITIES  WAITING   USEAGE    STUDENTS
            TIME                SERVICED
    9             0     37.22   172.00
    8             0     44.34   184.00
    7             0     43.92   158.00
    6           .18     48.02   171.00
    5          1.41     60.72   182.00
    4          4.39     79.98   169.00
    3         23.12     95.07   143.00
    2         82.38     99.05   108.00
    1        178.28     98.51    46.00

NUMBER OF   AVERAGE   AVERAGE   TOTAL
SERVICE     STUDENT   FACILITY  NUMBER
FACILITIES  WAITING   USEAGE    STUDENTS
            TIME                SERVICED
    9             0     35.07   179.00
    8             0     37.47   158.00
    7             0     45.88   183.00
    6           .16     48.48   177.00
    5           .83     69.21   182.00
    4          2.92     81.96   160.00
    3         20.89     92.13   150.00
    2         71.17     98.81   109.00
    1        185.23     99.30    58.00

NUMBER OF   AVERAGE   AVERAGE   TOTAL
SERVICE     STUDENT   FACILITY  NUMBER
FACILITIES  WAITING   USEAGE    STUDENTS
            TIME                SERVICED
    9             0     37.28   182.00
    8             0     42.47   163.00
    7             0     49.73   173.00
    6           .07     50.47   164.00
    5          1.74     66.38   167.00
    4          3.31     82.58   180.00
    3         44.85     98.00   144.00
    2         94.47     98.94    92.00
    1        175.99     99.62    60.00
```

Queue: Flowchart

Input:

```
    320  Read Arrival Rate - ARate                Students/Hour
              Service Rate - SRate                Min/Student
                Group Size - GRSize
         Number of Servers - NSER
          Hours in one Day - ETime
            Number of Days - Days
335-410  Print Input Data
    425  Convert Service Rate to Students/Hour          SRate = 60./SRate
    430  Convert Arrival Rate of Individual Students to
         Group Arrival Rate                             ARate = ARate/GRSize
         Return
```

Output:

```
465-485  Compute Cumulative Waiting Time - CWT
                     Total Groups Served - TOT
    490  Compute Average Waiting Time in Minutes - AVCWT
    495  Compute Total Students Served - TOT = TOT*GRSize
505-520  Compute Average Facility Useage, PST
522-560  IF First Line This Run, Print Headings
    565  Print Current Information
         Return
```

Queue: Flowchart

Main Program

```
 25- 40  Seed Random Number Generator
     45  Get Input Data
     55  Save Original Number of Servers in ITest
 65- 80  Set X, S Date Areas to Zero
110-145  Find Channel with Shortest Time to Availability
150-160  Test That Not All Channels Run to Maximum Time.  Otherwise Go to 250
165-175  Get A Sample Random Interarrival Time
180-210  Perform Service For Next Arrival on Selected Channel
215-220  Set Last Arrival Time Throughout Data Array
225-230  If Selected Channel Usage  7 Mx. Hours, Make it Unavailable
240-245  Step Cumulative Counters for Selected Channel
    250  Print Report of Results
    255  Reduce Number of Servers By 1
    260  # Servers > 0?
```

Queue: Flowchart

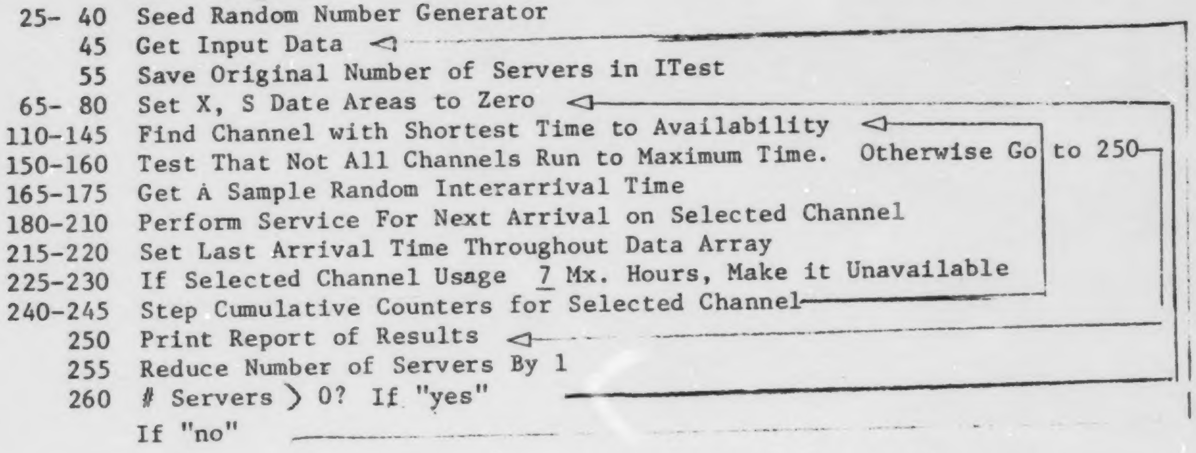
Input:

320 Read Arrival Rate - ARate — Students/Hour
Service Rate - SRate — Min/Student
Group Size - GRSize
Number of Servers - NSER
Hours in one Day - ETime
Number of Days - Days
335-410 Print Input Data
425 Convert Service Rate to Students/Hour — SRate = 60./SRate
430 Convert Arrival Rate of Individual Students to Group Arrival Rate — ARate = ARate/GRSize
Return

Output:

465-485 Compute Cumulative Waiting Time - CWT
Total Groups Served - TOT
490 Compute Average Waiting Time in Minutes - AVCWT
495 Compute Total Students Served - TOT = TOT*GRSize
505-520 Compute Average Facility Usage, PST
522-560 IF First Line This Run, Print Headings
565 Print Current Information
Return

Queue: Flowchart
Main Program



25- 40 Seed Random Number Generator
45 Get Input Data
55 Save Original Number of Servers in ITest
65- 80 Set X, S Date Areas to Zero
110-145 Find Channel with Shortest Time to Availability
150-160 Test That Not All Channels Run to Maximum Time. Otherwise Go to 250
165-175 Get A Sample Random Interarrival Time
180-210 Perform Service For Next Arrival on Selected Channel
215-220 Set Last Arrival Time Throughout Data Array
225-230 If Selected Channel Usage 7 Mx. Hours, Make it Unavailable
240-245 Step Cumulative Counters for Selected Channel
250 Print Report of Results
255 Reduce Number of Servers By 1
260 # Servers > 0? If "yes"
If "no"

Subroutine Raise

Subroutine Raise: Computes Exponential Random Variable Based on Number of Arrivals Per Unit Time or Number of Customers Serviced Per Unit Time.

```
     Subroutine Service:  For Selected Service Channel I, Next Event,
605       Establish Entrance Time into System of Next Event     x(I,2)
610       Is Channel then Busy?
615       Customer Waiting Time = 0                             x(I,3)
620       Put Channel in Service                                x(I,6)
625       Compute Channel Waiting Time                          x(I,5)
635       Compute Customer Waiting Time                         x(I,3)
640       Put Channel in Service When Free                      x(I,6)
645       Channel Waiting Time = 0                              x(I,5)
650       Get Random Service                                    x(I,4)
655       Compute Customer Exit Time From System                x(I,7)
660       Compute New Channel Available Time                    x(I,9)
665       Customer Total Time Customer in System (End Arrival   x(I,8)
                                                     Time)
          Return
```

Contents of Data Areas

State (50): 1 Thru Number of Servers Available
State (I) = 0 Means Server Not Yet Busy Full Time
= 1 Means Server Saturated, Not Available

X (50, 10) For Up to 25 Servers

	Odd: Next Group	Even: Channel
1	Random Interarrival Time (Groups)	Sum of Interarrival Time This Channel
2	Next Arrival Time	
3	Waiting Time, This Arrival	Sum of Waiting Times
4	Service Time, This Arrival	Sum of Facility Used Time
5	Server Idle Time, This Arrival	Sum of Idle Times
6	Entrance Time into Service	
7	Exit Time from Service (and System)	
8	Customer Total Time in System	Sum of Customer Used Time
9	Previous Exit Time from System	
10	Set to 1 for Channel in Use	Total Arrivals This Channel

X (I, 2) is Set Throughout Array for I Odd to Indicate Next Arrival Time for All Channels

```
5 PROGRAM QUEUE
6 DIMENSION X(50,10),STATE(50)
25 ISEED = 452368791
30 CALL RANFSET (ISEED)
35 ISV1 = ISEED
40 ISV2 = ISEED + 45689321
45 CALL INPUT (ARATE,SRATE,GRSIZE,NSER,ETIME,DAYS)
55 ITEST = NSER
60 DAYCT = 1.0
65 DO 80  I=1,50
70 STATE(I)=0.0
75 DO 80 J=1,10
80 X(I,J)=0.0
85 N = 2*NSER
90 ANT=100.0
95 NT=ANT
100 DO 248  I4=1,1000
101 K=I4
110 SHORT=99999.9
115 DO 145  I=1,N,2
120 IT=I/2+1
125 IF (STATE(IT) .EQ. 1.0) GO TO 145
130 IF (SHORT-X(I,7))  145,135,135
135 ISAVE = I
140 SHORT=X(I,7)
145 CONTINUE
150 DO 155 II=1,NSER
155 IF (STATE(II) .NE. 1.0) GO TO 165
160 GO TO 250
165 CALL RANFSET(ISV1)
170 CALL RAISE (RIT,ARATE)
175 CALL RANFGET(ISV1)
180 I = ISAVE
185 X(I,1)=RIT
190 X(I,10)=1.0
195 CALL RANFSET(ISV2)
200 CALL SERV (X,I,SRATE)
210 CALL RANFGET(ISV2)
215 DO 220  II=1,N,2
220 X(II,2)=X(I,2)
225 IT=I/2+1
230 IF (X(I,9) .GE. ETIME) STATE(IT)=1.0
235 II=I
240 DO 245 J=1,10
245 X(II+1,J)=X(II+1,J)+X(II,J)
248 CONTINUE
250 CALL REPORT (ITEST,X,N,NSER,GRSIZE)
255 NSER=NSER-1
256 CALL RANFSET (ISEED)
260 IF (NSER .GT. 0)  GO TO 65
265 ISEED = ISEED + 254689321
270 GO TO 45
275 END
```

```
300 SUBROUTINE INPUT (ARATE,SRATE,GRSIZE,NSER,ETIME,DAYS)
304 DIMENSION X(50,10),STATE(50)
310 PRINT 311
311 FORMAT (1H1)
320 READ, ARATE,SRATE,GRSIZE,NSER,ETIME,DAYS
325 IF (NSER .EQ. 0)  STOP
335 PRINT 340, ARATE
340 FORMAT (*ARRIVAL RATE,STUDENTS PER HOUR=*,F6.2)
350 PRINT 355,SRATE
355 FORMAT (*SERVICE RATE,MINUTES PER STUDENT=*,F6.2)
365 PRINT 370, GRSIZE
370 FORMAT (*GROUP SIZE=*,F6.2)
380 PRINT 385, NSER
385 FORMAT (*STARTING NUMBER OF SERVICE FACILITIES=*,I3)
395 PRINT 400, ETIME
400 FORMAT (*LENGTH OF DAY IN HOURS=*,F6.2)
410 PRINT 415, DAYS
415 FORMAT (*NUMBER OF DAYS IN SIMULATION=*,F6.2)
425 SRATE=60.0/SRATE
430 ARATE=ARATE/GRSIZE
435 RETURN
440 END
450 SUBROUTINE REPORT (ITEST,X,N,NSER,GRSIZE)
454 DIMENSION X(50,10)
465 CWT=0.0
470 TOT=0.0
475 DO 485  I=2,N,2
480 CWT=CWT+X(I,3)
485 TOT=TOT+X(I,10)
490 AVCWT=CWT/TOT*60.0
495 TOT=TOT*GRSIZE
500 PST=0.0
505 DO 510 I=2,N,2
510 PST=X(I,4)/X(I-1,7)*100.0+PST
515 XS=NSER
520 PST=PST/XS
522 IF (ITEST .NE. NSER) GO TO 565
525 PRINT 530
530 FORMAT (1H1)
540 PRINT 541
541 FORMAT (*NUMBER OF*,2X,*AVERAGE*,2X,*AVERAGE*,2X,*TOTAL*)
545 PRINT 546
546 FORMAT (*SERVICE*,4X,*STUDENT*,2X,*FACILITY*,1X,*NUMBER*)
550 PRINT 555
555 FORMAT (*FACILITIES*,1X,*WAITING*,2X,*USEAGE*,3X,*STUDENTS*)
559 PRINT 560
560 FORMAT (12X,*TIME*,14X,*SERVICED*)
565 PRINT 570, NSER,AVCWT,PST,TOT
570 FORMAT (I6,F12.2,2F9.2)
575 RETURN
580 END
```

```
600 SUBROUTINE SERV (X,I,SRATE)
603 DIMENSION X(50,10)
605 X(I,2)=X(I,2)+X(I,1)
610 IF (X(I,2)-X(I,9))  635,635,615
615 X(I,3)=0.0
620 X(I,6)=X(I,2)
625 X(I,5)=X(I,6)-X(I,9)
630 GO TO 650
635 X(I,3)=X(I,9)-X(I,2)
640 X(I,6)=X(I,9)
645 X(I,5)=0.0
650 CALL RAISE (X(I,4),SRATE)
655 X(I,7)=X(I,6)+X(I,4)
660 X(I,9)=X(I,7)
665 X(I,8)=X(I,7)-X(I,2)
670 RETURN
675 END
680 SUBROUTINE RAISE (ERN,CR)
685 AVIT=1.0/CR
690 R=RANF(-1)
695 ERN=-AVIT*LOGF(R)
700 RETURN
705 END
715 ENDPROG
900 20.0,10.0,1.0,20,9.0,1.0
901 0.0,0.0,0.0,0,0.0,0.0
```

ADULT RESOURCES FLOW (ARF-1)

The Adult Resources Flow Model (ARF) was created as a tool to help explore the relationships between intended staff availability and the anticipated work to be done in a school situation. The staff is described in terms of type of person, number of each type, cost of each type, hours each type is available, and a description of the work each is to do in per cent of time devoted to each intended task for that type. Each kind of work is described in terms of range and distribution of time for each occurrence, and occurrences as a function of number of students, number of staff members, and week in the school cycle. The Model is available in both a batch and time sharing version. The batch version of the Model will run for 20 weeks while the time sharing version runs one week at a time.

The Model creates, within the specified parameters, the sample workload for each kind of work for each week. This load may be contrasted with intended staff availability.

Several levels of analysis may be made:

1. If the total hours required by the sample workload exceeds the total of staff hours available, then either:
    a. A way has to be found to cut down the amount or duration of the work that needs to be done, or
    b. More staff time is needed. In this case:
        1) The number of staff members may be increased, and/or
        2) Staff members may be asked to work longer hours.

2. If the hours needed for a particular type of work exceed the staff hours available for that type of work, then:
    a. The requirements for that type of work may be reduced, or
    b. The allocation of time of staff competent to perform that type of work can be changed so that these staff members do less of something else, or
    c. Other types of staff members not originally intended to do the work in question can be allocated to it, or
    d. More staff members of a given type can be made available.

3. The input data represents the best guesses of project planners. If the ARF Model does not show a reasonable correspondence between staff and work, more thinking can be done about the nature of the educational experiences being planned, and about different ways of accomplishing the intended objectives.

The major usefulness of the ARF Model has been to make certain assumptions about a planned educational program and its staffing pattern more explicit. A secondary use, and the intended reason for creating the model, has been to explore staffing patterns in relationship to the work implied in an educational program being planned. A third use has been to define work sampling and time study research that might be done to provide better parameters for a model such as this. Such research has merits also in helping provide insight into how educational systems work at the human interaction level.

Experiences with the ARF Model has indicated two potential problems to be considered, aside from the validity of input data. One is the inclusion of work such as "lunch" that should be uniformly distributed among all people and at rather limited times. Total hour requirements and availability for such work is not as much of a guarantee of a "match", assuming the figures were about the same, as would be the case with both time flexibility and staff member flexibility. Two lunches at 4:00 p.m. for one staff member doesn't quite compensate for no lunch for another staff member or for not having lunch at noon! Thus, lunch is not a random phenomenon.

A similar problem occurs if students replace staff in the ARF Model. It may not make much difference which staff member gives a test, but it usually makes a lot of difference which student takes it. Here, again, a match between hour requirements and hour availability is much less reassuring than if the work were interchangeable among staff members.

If both staff and students (as students, not as helpers) are included in the same computer run, there is a potential problem in interpreting the staff and student roles, which may not be really interchangeable. More student time available for an activity such as planning may not compensate for less staff time; more student need for such activities may even require more staff time. Both staff and students may need to be involved simultaneously, and the role of each may be quite different.

Consequently, the recommended use of the ARF Model remains, as intended, to match staff availability with work requirements. If this model is used for students at all, it should be used for students alone, and with the cautions noted for interpreting the results. The EDSIM models have been designed to investigate student activities in a planned educational system.

The following pages present a sample problem, showing the computer output available from ARF-1 batch session. This problem shows that trial data from the Fort Lincoln New Town Project in Washington D.C.; as with all such data, is not to be taken as representative of actual plans for that project. Similar runs have been made for some Centers at the University of Massachusetts, School of Education.

Figure 1 displays all the staff data except work description. For this run, 35 regular and 50 student helper staff members provide 1900 hours of service per week at a total annual cost of $341,600. No overtime work was included, but regular staff members could work 40 hours per week (which is longer than most schools require their staff members to be on-site).

Figure 2 describes the work anticipated for each type of staff member. It is read as follows: For staff type 1 (teacher stage 2-4 level 2), priority 2 is work type 1 (counseling,prescribing) for about 20% of the staff member's time. Priority 1 is work type 2 (planning), no less than 10 per cent of the time, but no

```
                SUMMARY OF INPUT INFORMATION FOR STAFF WORK, FLNT, 8-1

      STUDENTS= 700, FACULTY= 35

                                                                      *STAFF*

      TYPE          TYPE NAME                               NUMBER  COST  TOTAL COST HOURS OVER

         1  TEACHER STAGE 2-4 LEVEL 2                           2  14900     29800      40     0

         2  TEACHER STAGE 1   LEVEL 2                           1  14900     14900      40     0

         3  TEACHER COORDINATING LEVEL 3                        6  12500     75000      40     0

         4  TEACHER NURSING    LEVEL 3                          1  12500     12500      40     0

         5  TEACHER RESOURCE CENTER LEVEL 3                     1  12500     12500      40     0

         6  TEACHER GRAPHIC ARTS LEVEL 3                        1  12500     12500      40     0

         7  TEACHER SHOP  LEVEL 3                               1  12500     12500      40     0

         8  TEACHER AV   LEVEL 4                                1   8850      8850      40     0

         9  TEACHER MUSIC LEVEL 4                               1   8850      8850      40     0

        10  TEACHER PHYS ED LEVEL 4                             2   8850     17700      40     0

        11  TEACHER         LEVEL 4                             6   8850     53100      40     0

        12  TEACHER-AIDE -                                     12   5700     68400      40     0

        13  STUDENT HELPER - TUTOR                             50    300     15000      10     0

            ***TOTALS                                          85       $ 341600

      *TOTAL HOURS AVAILABLE=  1900 REG +     0 OVER =  1900
```

Figure 1 - Staff Description

STAFF WORK DESCRIPTION

FIGURE 2

```
                STAFF WORK ASSIGNMENT FOR STAFF WORK, FLNT, 8-1
TY  PR TY MI MA   PR TY MI MA   PR TY MI MA   PR TY MI MA   PR TY MI MA   PR TY
 1 * 2  1 20- 0 *  1  2 10-20 *  3  3 25- 0 *  4  4 20- 0 *  5  8  1- 3 *  6  9
 2 * 2  1 20- 0 *  1  2 10-20 *  3  3 25- 0 *  4  4 20- 0 *  5  8  1- 3 *  6  9
 3 * 1  1 20- 0 *  2  2 10-20 *  3  3 25- 0 *  4  4 20- 0 *  5  8  1- 3 *  7  9
 4 * 1  1 20- 0 *  4  2  5-10 *  2  3 22- 0 *  3  4 20- 0 *  7  8  1- 3 *  5  9
 5 * 5  1 20- 0 *  4  2  5-10 *  1  3 15- 0 *  3  4 10- 0 *  7  8  1- 3 *  9  9
 6 * 4  1 20- 0 *  5  2  5-10 *  1  3  9- 0 *  2  4 13- 0 *  7  8  1- 3 *  8  9
 7 * 3  1 20- 0 *  4  2  5-10 *  1  3 22- 0 *  2  4 20- 0 *  7  8  1- 3 *  8  9
 8 * 3  1 20- 0 *  5  2  5-10 *  1  3 10- 0 *  2  4 20- 0 *  7  8  1- 3 *  8  9
 9 * 3  1 20- 0 *  4  2  5-10 *  2  3 14- 0 *  1  4 33- 0 *  6  8  1- 3 *  7  9
10 * 4  1 20- 0 *  3  2  5-10 *  2  3  9- 0 *  1  4 40- 0 *  6  8  1- 3 *  7  9
11 * 3  1 20- 0 *  5  2  5-10 *  1  3 24- 0 *  2  4 16- 0 *  7  9  5- 0 *  4 10
12 * 9  2  0- 5 *  1  3 22- 0 *  7  8  1- 3 *  6  9  5- 0 *  2 10 20- 0 *  3 11
13 * 7  2  2- 6 *  5  3 15- 0 *  0  0  0- 0 *  0  0  0- 0 *  0  0  0- 0 *  4 10
```

more than 20 per cent, etc. If only a minimum time (MI) is given, this is treated as an average. If both a minimum and a maximum time (MA) is given, the intent is to provide stricter limits. Had any of the work assignments added up to more than 100% of a staff member's time, an "error note" to this effect would have been printed. Figure 2 also indicates that work type 1 is highest priority for staff types 3 and 4, and lowest priority for staff types 12 and 13.

Figure 3 summarizes the staff hours available for various types of work, from what are, in effect, the job descriptions presented in Figure 2. Regular hours are calculated as the minimum per cent times the hours available times the number of staff members; extra regular hours are calculated as the difference between maximum and minimum per cent, where both are given, times the hours available times the number of staff members. Because of typical leeway in describing a staff job, the hours indicated in this table may add up to more than the total hours available as shown in Figure 1. If this should occur, it means that once some of the regular extra hours have been used, other allocations in this category cannot be used. The data shown adds up to less than the 1900 hours available, and hence all indicated hours may be utilized.

In effect, Figure 3 indicates a range of hours available for a work activity, if job descriptions are followed. For work type 2, planning, 74 hours per week are regularly available, with 108 more hours available (or a total of 182 hours) if the upper limits of staff availability are used. These hours represent all staff members, not just those for whom this type of work is a high priority. However, if a staff member does not have this type of work in his job description at all, his availability does not show in the hours available for this type of work.

Figure 3

Summary of Staff Hours Available

```
        HOURS AVAILABLE FOR TYPE OF WORK,      STAFF WORK, FLNT, 8-1

TYPE   TITLE OF TYPE OF WORK         REGULAR    OVER TIME      EXTRA HOURS
                                      HOURS       HOURS     REGULAR      OVER
  1    COUNSELING, PRESCRIBING       184.0        0.0         0,0        0,0
  2    PLANNING                       74.0        0.0       108,0        0,0
  3    TUTORING                      372.2        0.0         0,0        0,0
  4    GROUP INSTRUCTION             188.8        0.0         0,0        0,0
  8    PARENTS + COMMUNITY            11.6        0.0        23,2        0,0
  9    IN SERVICE TRAINING            70.0        0.0         0,0        0,0
 10    TESTING                       152.2        0.0        26,6        0,0
 11    SCORING/RECORDING             111.0        0.0        10,0        0,0
 12    MATERIALS PREPARATION          42.0        0.0        84,4        0,0
 13    SUPPORTING DUTIES              86.0        0.0        98,0        0,0
 14    DISCIPLINE                      0.0        0.0        18,0        0,0
 15    SUPERVISING ACADEMIC ACT      133.0        0.0        57,0        0,0
                                   ------------------------------------------
     ***TOTALS--                    1424.8        0.0       425,2        0,0
```

Figure 4 summarizes the input data for work occurrences. For example, work type 8, Parents and community work, has a relative priority of 4 (not currently used by ARF-1 for calculations), time distribution type 3 (exponential decay; most occurrences will be towards the short end of the range of time), a time range of .1 to .3 hours (6 to 18 minutes), and will occur, on the average, .1 times per student per week plus 1.0 times per faculty member per week. For the indicated 700 students and 35 faculty members, then, a normal expectation of such work occurrences would be 105 times per week. Had a constant term been shown, this would have been added to the occurrences due to students and faculty. In effect, this is the number of times the appropriate time distribution will be sampled to get the total weekly hour demand for an average week.

But not all weeks are average. For work type 8 in Figure 4, week 9 is 1 unit above average, week 10 is 2 units above average (average is indicated by 5 in the profile by week; 4 indicates 1 unit more; 6, 1 unit less, etc.). The per cent change column indicates that 1 unit in the profile is worth a 300 per cent change. Consequently, week 9 will have not 105 occurrences, but 420 occurrences.

Figure 5 shows, for each type of work, the calculated occurrences each week, and the result of sampling the indicated time distribution that number of times as the hours needed to do that much work. Following work type 8, an average week had 105 occurrences (OC), taking between 13.9 and 15.4 hours (HR), while an extreme week with 735 occurrences might take 100.7 hours of staff time.

Figure 5 also shows the hours total per week, which in all cases is less than the 1900 hours available. Hence, all the required work can get done (assuming the work description parameters represent reality), although some staff job description changes may appear desirable.

As an example, work type 8, requiring 100.7 hours in an extreme week, has a maximum of 34.8 hours available for it from the staff job descriptions. Were this an actual, operating school situation, the decision would have to reassign staff appropriately. By inexpensive computer simulation runs, the effects of possible staff reassignments may be explored.

The following pages give a summary of the input data, card layouts, operating instructions for the computer program, and the program listing.

```
WORK  REL   DIST               LOAD  FACTORS         PCT    PROFILES                                                              TITLE O
TYPE  PRI   TYPE     RANGE    STU    FAC    CONST CHANGE          BY WEEK                                                          WORK

 1     1     3′     .1-0.2   2,75   0.00   0,00   0.00    5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5    COUNSEL
 2     2     3      .5-1.0   0,00   5.00   0,00   0.50    3 3 4 5 5 5 5 5 4 4 5 5 5 5 5 4 4 5 5 5    PLANNIN
 3     2     3      .1-0.3   2,44   0.00   0,00   0.50    6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5    TUTORIN
 4     5     1      .4-0.6   0,10   0.00   0,00   0.50    6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5    GROUP I
 8     4     3      .1-0.3   0,10   1.00   0,00   3.00    5 5 5 5 5 5 5 5 4 3 3 4 5 5 5 5 5 5 4 3    PARENTS
 9     5     1      .4-1.0   0,00   5.00   0,00   0.00    5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5    IN SERV
10     2     3      .4-0.8   0,40   0.50   0,00   0.30    4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5    TESTING
11     3     2      .0-0.1   1,00   0.00   0,00   0.30    4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5    SCORING
12     6     3      .5-1.0   0,00   1.00   0,00   1.00    4 4 4 5 5 5 5 4 4 4 5 5 5 5 5 4 4 5 5 5    MATERIA
13     7     3      .5-1.5   0,00   5.00   0,00   0.50    3 4 4 5 5 5 5 4 3 4 5 5 5 5 5 4 3 4 5 5    SUPPORT
14     9     1      .1-0.1   0,20   4.00   0,00   0.00    5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5    DISCIPL
15     2     1      .8-1.5   0,00   3.00   0,00   0.50    5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5    SUPERVIS
*                *                *                *                *                *                *


               DISTRIBUTION TYPES

                    1 = UNIFORM
                    2 = NORMAL
                    3 = EXPONENTIAL
```

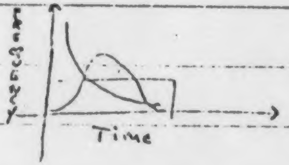


Figure 4

```
         DEMAND FOR TYPE OF WORK,        STAFF WORK, FLNT, 8-1

 1      2      3      4      5      6      7      8      9      10     11     12     13
YPE


OUNSELING, PRESCRIBING
925   1925   1925   1925   1925   1925   1925   1925   1925   1925   1925   1925   192
1.  235.5 229,1 233,0 235.8 226,8 229,5 228,3 227.4 229,0 223,7 230.2 229,

LANNING
35.    350    262    175    175    175    175    175    262    262    175    175    17
6,9 267.5 157,6 1 3,6 104.0 105,8 1 5,3 104,9 155,9 157,0 105,2 103,8 105,

UTORING
854    854   1706   1708   1708   1708   1708   1708   1708   1708   1708   1708   170
2.1 127.0 247,9 234.5 233.4 245,9 236.0 241.4 237,9 232,9 234,6 231,9 235,

ROUP INSTRUCTION
 35     35     70     70     70     70     70     70     70     70     70     70     7
1.1   20.8   43.1   40.4   42.0   41,    43.1   42,8   42.0   41,5   42,4   41,8   40,

ARENTS + COMMUNITY
105    105    105    105    105    105    105    105    420    735    735    420    10
3.9   14.4   14.5   15.4   14.4   14.5   14.7   14.7   61.3  100,7  100,7   57.1   14,

N SERVICE TRAINING
175    175    175    175    175    175    175    175    175    175    175    175    17
2.6 122.2 125,7 125.7 121.1 122,1 122.0 121.0 123.4 116,4 122,6 122,6 121,

ESTING
386    386    297    297    297    297    297    297    297    297    297    297    29
9.2 197.3 143.5 142.6 143.9 142,6 143.0 144.1 143,7 144,6 144,9 142,7 139,

CORING/RECORDING
91     910    70     700    700    700    700    700    700    700    700    700    70
4,6   42.1   37,9   40.2   40.0   34.2   38.2   34.5   31,7   31,8   29,2   32,4   26,5

ATERIALS PREPARATION
7.     70     70     35     35     35     35     70     70     70     35     35     35
1,6    1.4    1.2    0.8    0.7    0,8    0.6    1.4    1,5    1,4    0,7    0,6    0,9

UPPORTING DUTIES
35.    262    262    175    175    175    175    262    350    262    175    175    175
3.8 184.6 18 .  125,6 120.4 119,8 123.3 182.7 235.1 182,7 115.8 124.2 125,2

ISCIPLINE
28.    280    280    280    280    280    280    280    280    280    280    280    280
 .    28.0   28,    28.0   28.0   28,    28.0   28.0   28.0   28,0   28,0   28,0   28,0

UPERVISING ACADEMIC ACT
105    105    105    105    105    105    105    105    105    105    105    105    105
 ,8 119.7 12 .3 119,8 121.3 124,  123.1 121.3 120.3 122,5 124.2 119,8 122,5
          *                *                *                *                *
LS
 ,6        1328,9        1205,0        1276,8        1408,2        1272,0        1190,6
    1360.3        1209.6        1205,6        1265.1        1368,4        1235,1
```

FIGURE 5

SIMULATED DEMAND FOR TYPE OF WORK BY WEEK

Card Layouts

Title Card

Col.	Contains
1- 4	Number of Students
5- 8	Number of staff
9-10	Number of staff descriptor cards to be read (sets)
11-12	Number of work descriptor cards to be read
13-20	unused
21-44	Title (for page headings)
45-53	Initial setting for random number generator

Staff Descriptor card

Col.	Contains
1- 2	Type number
3- 5	Number of staff members of this type
6-10	Cost of 1 staff member of this type (no decimal point)
11-12	Regular hours work for 1 staff member of this type (week)
13-14	Overtime hours work (week)
15	Priority for this type of staff member of first work type
16-17	First work type for this staff type
18-19	Average or minimum percent of time for this work (needed)
20-21	Maximum percent for this work (optional)

Provision for 8 more work assignments, same format, starting in col. 22, 29, 36, 43, 50, 57, 64, 71

Staff name card

Col.	Contains
1- 2	Staff type number
3	unused
4-39	Description (name of staff type)

Each of these cards follow the staff descriptor card for the same type of staff.

Work type card

Col.	Contains
1- 2	Type number
3	Relative priority of this type of work (1 thru 9)
4	Distribution type for this work (1 = Uniform, 2 = Normal, 3 = Exponential)
5- 6	Minimum time for an occurrence of this work (x.x hours)
7- 8	Maximum time for an occurrence of this work (x.x hours)
9-11	Student load factor for this work (x.xx)
12-14	Faculty load factor for this work (x.xx)
15-17	Constant load factor for this work (x.xx)
18-19	Percent change corresponding to 1 unit in week descriptor
20	unused
21-40	20 week descriptors. (5 = normal, or load as calculated. 4 = 1 above [1 percent change unit] 6 = 1 below)
41-55	unused
56-79	label (name) for this work type

Set Up of Source Decks

1. Set up the source or object program deck according to the computer and operating system being used.
2. Add a jobs card with Col. 1-3 = number of data sets to follow.
3. Add the number of data sets specified in the jobs card. Each data set consists of the following:
    a. Title card
    b. The number of staff descriptor/name card sets specified in the title card. Each set consists of:
        1) Staff descriptor card
        2) Staff name card associated with preceding descriptor card
4. If the computer or operating system being used requires an end of job or end of file card, add it.

Program Notes:

The following program listing is the CDC 3600 version used at the University of Massachusetts. The program structure was designed to provide for tabulation of individual occurrences; ARF-1 uses totals only, and hence the program as listed samples a distribution of the mean rather than the distribution of individual occurrences.

```
      PROGRAM APE
      DIMENSION WORK(16,20),Q(15),ITOTL(15)
      DIMENSION NTYP(15),NCOST(15),HRS(15),OVHR(15),IP(15,9),IWT(15,9),
     1PCA(15,9),PCB(15,9)
      DIMENSION IWTP(15),IDTY(15),PA(15),PB(15),ST(15),FA(15),WLD(15),
     1PCT(15),PFL(15,20)
      DIMENSION TITLE(4),IA(15,9),IB(15,9)
      DIMENSION NB(20),TALEW(15,4),TALEP(15,6)
      COMMON Q,WORK,NTYP,NCOST,HRS,OVHR,IP,IWT,PCA,PCB,IWTP,IDTY,PA,PB,S
     1T,FA,WLD,PCT,PFL,TITLE,ITOTL
      COMMON NB,TALEW,TALEP,IB,IA
      INTEGER TITLE,TALEW,TALEP
      READ 900, NJOB
  900 FORMAT (I3)
      DO 901  KJOB=1,NJOB
      DO 21 L=1,15
      DO 21 LL=1,4
   21 TALEW(L,LL)=0
      DO 22 L=1,15
      DO 22 L1=1,6
   22 TALEP(L,L1)=0
      DO 20 I=1,15
      NTYP(I)=0
      NCOST(I)=0
      HRS(I)=0.0
      OVHR(I)=0.0
      IWTP(I)=0
      IDTY(I)=0
      ITOTL(I)=0
      IDTY(I)=0
      PA(I)=0.0
      PB(I)=0.0
      ST(I)=0.0
      FA(I)=0.0
      WLD(I)=0.0
      PCT(I)=0.0
      DO 18 J=1,9
      IP(I,J)=0
      IWT(I,J)=0
      IA(I,J)=0
      IB(I,J)=0
      PCA(I,J)=0.0
   18 PCB(I,J)=0.0
      DO 20 J=1,20
   20 PFL(I,J)=0.0
      READ 2,STU,FAC,I,J,TITLE,SET
    2 FORMAT(2F4.0,2I2,8X,4A6,F9.0)
      IF(SET)5,5,400
    5 Q7=167.
      SET=TIMEF(Q7)
    6 Y=RANFSET(SET)
  400 PRINT 401, SET,TITLE
  401 FORMAT (1H1,1X,*RANDOM NUMBER SET TO=*,F20.9,20X,5HFOR   ,4A6,//)
    3 READ 4,K,NTYP(K),NCOST(K),HRS(K),OVHR(K),(IP(K,L),IWT(K,L),PCA(K,L
     1),PCB(K,L),L=1,9)
  4 FORMAT(I2,I3,I5,2F2.0,9(I1,I2,2F2.2))
    READ 5,K,(TALEP(K,KK),KK=1,6)
  5 FORMAT(I2,1X,6A6)
    I=I-1
    IF(I)7,7,3
  7 DO 14 I=1,15
    IF(NTYP(I))14,14,9
  9 TOT=0.0
    DO 11 L=1,9
 11 TOT=TOT+PCA(I,L)
    IF(TOT-1.0))14,12,12
 12 PRINT 13,I,(TALEP(I,KK),KK=1,6),TOT,TITLE
 13 FORMAT(16H0FOR PERSON TYPE,I3,2H (,6A6,1H),1X,13HPERCENT WORK=,F5.
   12,1X,4A6)
 14 CONTINUE
    THRAV=0.0
    NTTYP=0
    ITOT=0
    THRS=0.0
    TOVHR=0.0
 24 READ 25,K,IWTP(K),IDTY(K),PA(K),PB(K),ST(K),FA(K),WLD(K),PCT(K),
   1(PFL(K,L),L=1,20),(TALEW(K,KK),KK=1,4)
 25 FORMAT(I2,2I1,2F2.1,3F3.2,F2.2,1X,20F1.0,15X,4A6)
    IF (PA(K) - PB(K)) 903,903,902
902 PJ = PA(K)
    PA(K) = PB(K)
    PB(K) = PJ
903 J=J-1
    IF(J)28,28,24
 28 PRINT 29,TITLE
 29 FORMAT(1H1,23X,33HSUMMARY OF INPUT INFORMATION FOR ,4A6,//)
    PRINT 31,STU,FAC
 31 FORMAT(1H0,9HSTUDENTS=,F4.0,10H, FACULTY=,F4.0,//,55X,7H*STAFF*,//
   1)
    PRINT 34
 34 FORMAT(3X,4HTYPE,10X,9HTYPE NAME,22X,6HNUMBER,3X,4HCOST,2X,10HTOTA
   2L COST,2X,5HHOURS,2X,4HOVER,/)
    DO 40 K=1,15
    IF(NTYP(K))40,40,37
 37 ITOTL(K)=NTYP(K)*NCOST(K)
    X=NTYP(K)
    TOVHR=TOVHR+OVHR(K)*X
    NTTYP=NTTYP+NTYP(K)
    ITOT=ITOT+ITOTL(K)
    THRS=THRS+HRS(K)*X
    DO 35 L=1,9
    IA(K,L)=PCA(K,L)*100.
 35 IB(K,L)=PCB(K,L)*100.
    PRINT 36,K,(TALEP(K,KK),KK=1,6),NTYP(K),NCOST(K),ITOTL(K),HRS(K),
   3OVHR(K)
 36 FORMAT(/,4X,I2,3X,6A6,4X,I3,4X,I5,2X,I7,7X,F2.0,4X,F2.0)
 40 CONTINUE
    PRINT 38,NTTYP,ITOT
 38 FORMAT(/,9X,9H***TOTALS,30X,I4,10X,1H$,I7)
```

```
    THRAV=THRS+TOVHR
    PRINT 39,THRS,TOVHR,THRAV
 39 FORMAT(/,2X,24H**TOTAL HOURS AVAILABLE=,F6.0,6H REG +,F6.0,7H OVER
   1 =,F6.0)
    PRINT 180,TITLE
180 FORMAT(1H1,40X,25HSTAFF WORK ASSIGNMENT FOR,1X,4A6,//,1X,2HTY,
   1    9(3X,2HPR,1X,2HTY,1X,2HMI,1X,2HMA))
    DO 185 K=1,15
    IF(IP(K,1))185,185,184
184 PRINT 186,K,(IP(K,L),IWT(K,L),IA(K,L),IB(K,L),L=1,9)
186 FORMAT(/,1X,I2,9(1X,1H*,3(I3),1H-,I2))
185 CONTINUE
    PRINT 42,TITLE
 42 FORMAT(1H1,15X,33HHOURS AVAILABLE FOR TYPE OF WORK,,6X,4A6)
    PRINT 44
 44 FORMAT(1H0,4HTYPE,3X,21HTITLE OF TYPE OF WORK,8X,7HREGULAR,5X,9HOV
   1ER TIME,6X,11HEXTRA HOURS,/,38X,5HHOURS,8X,5HHOURS,3X,7HREGULAR,4X
   2,5H OVER)
     TTL1=0.0
     TTL2=0.0
     TTL3=0.0
     TTL4=0.0
     DO 56 K=1,15
     THRS=0.0
     TOVHR=0.0
     THRSX=0.0
     TOVHRX=0.0
     DO 52 L=1,15
     X=NTYP(L)
     DO 52 M=1,9
     IF(IWT(L,M)-K)52,49,52
 49  THRS=THRS+PCA(L,M)*HRS(L)*X
     TOVHR=TOVHR+PCA(L,M)*OVHR(L)*X
     XL=PCB(L,M)-PCA(L,M)
     IF (XL)  52,52,404
404  THRSX=THRSX+XL*HRS(L)*X
     TOVHRX=TOVHRX+XL*OVHR(L)*X
 52  CONTINUE
     IF (THRS+THRSX)  56,56,54
 54  PRINT 55,K,(TALEW(K,KK),KK=1,4),THRS,TOVHR,THRSX,TOVHRX
 55  FORMAT(1X,I3,4X,4A6,1X,4F10.1)
     TTL1=TTL1+THRS
     TTL2=TTL2+TOVHR
     TTL3=TTL3+THRSX
     TTL4=TTL4+TOVHRX
 56  CONTINUE
     PRINT 57
 57  FORMAT(37X,49H-------------------------------------------------)
     PRINT 58,TTL1,TTL2,TTL3,TTL4
 58  FORMAT(6X,11H***TOTALS--,16X,4F10.1)
     PRINT 108,TITLE
108  FORMAT(1H1,25X,31HDISTRIBUTIONS OF WORK TYPES FOR,1X,4A6,//)
```

```
    PRINT 110
110 FORMAT(1X,4HWORK,2X,3HPFL,3X,4HDIST,15X,13HLOAD  FACTORS,5X,3HPCT,
   13X,8HPROFILES,36X,8HTITLE OF)
    PRINT 125
125 FORMAT(1X,4HTYPE,2X,3HPRI,3X,4HTYPE,4X,5HRANGE,4X,3HSTU,4X,3HFAC,3
   2X,5HCONST,1X,6HCHANGE,6X,7HBY WEEK,35X,9HWORK TYPE,/)
    DO 145 KQ=1,15
    IF(IDTY(KQ))145,145,130
130 PRINT 140,KQ,IWTP(KQ),IDTY(KQ),PA(KQ),PB(KQ),ST(KQ),FA(KQ),WLD(KQ)
   1,PCT(KQ),(PFL(KQ,L),L=1,20),(TALEW(KQ,LQ),LQ=1,4)
140 FORMAT(/,1X,I2,    2(4X,I2),4X,F3.1,1H-,F3.1,3(2X,F5.2),3X,F4.2,2X,
   120F2.0,4X,4A6)
145 CONTINUE
    PRINT 155
155 FORMAT(1X,1H*,8(15X,1H*))
    PRINT 170
170 FORMAT(////,26X,18HDISTRIBUTION TYPES,//,30X,*1 = UNIFORM*,/,30X,
   5*2 = NORMAL*,/,30X,*3 = EXPONENTIAL*)
    PRINT 60,TITLE
 60 FORMAT(1H1,24X,24HDEMAND FOR TYPE OF WORK.,6X,4A6,//)
    DO 63 K=1,16
    DO 63 L=1,20
 63 WORK(K,L)=0.0
    PRINT 82
 82 FORMAT(1X,*WEEK   1      2      3      4      5      6      7      8
   19     10     11     12     13     14     15     16     17     18     19     2
   20  TOTAL*,//,1X,9HWORK TYPE,/)
    DO 80 K=1,15
    TTOT=0.0
    IF (IDTY(K))80,80,66
 66 X=ST(K)*STU+FA(K)*FAC+WLD(K)
    DO 79 L=1,20
    N=X*(1.+(5.-PFL(K,L))*PCT(K))
    NP(L)=N
    IF(N-300)69,69,68
 68 A=  PA(K)
    B=  PB(K)
    M=IDTY(K)
    GO TO(210,250,275)    ,M
210 EX = (B+A)/2.0
    VARX = ((B-A)**2)/12.0
    GO TO 280
250 EX = (B+A)/2.0
    VARX=(B-A)/6.0
    GO TO 280
275 EX =((B-A)*.20 +A)
    VARX = EX**2
280 EX = N*EX
    VARX  = N* VARX
    SIGMA = SQRTF(VARX)
    CALL NDIST(Y,EX,SIGMA)
    WORK(K,L)=WORK(K,L)+Y
    GO TO 81
 69 IF (N) 79,79,70
 70 M=IDTY(K)
    DO 78 MM=1,N
    GO TO (73,75,77),M
 73 CALL UNDIST(Y,PA(K),PB(K))
    GO TO 78
 75 A=  PA(K)
    B=  PB(K)
    EX = (B+A)/2.0
    SIGMA=(B-A)/6.
    CALL NDIST(Y,EX,SIGMA)
    GO TO 78
 77 EX=((B-A)*.200)
    CALL EXPEXT(EX,Y)
    Y=Y+A
 78 WORK(K,L)=WORK(K,L)+Y
 81 WORK(16,L)=WORK(16,L)+WORK(K,L)
 86 TTOT=TTOT+WORK(K,L)
 79 CONTINUE
    PRINT 87,K,(TALEW(K,KK),KK=1,4)
 87 FORMAT(/,1X,I2,3X,4A6)
    PRINT 85,NB
 85 FORMAT(1X,2HOC,1X,20I6)
    IF(TTOT)90,90,88
 88 PRINT 89,(WORK(K,L),L=1,20),TTOT
 89 FORMAT(1X,2HHR,1X,20F6.1,F8.1)
 90 CONTINUE
 80 CONTINUE
    PRINT 100
100 FORMAT(1X,1H*,8(15X,1H*),/,1X,9H***TOTALS)
    PRINT 92,(WORK(16,L),L=1,20,2)
 92 FORMAT(F10.1,9F12.1)
    PRINT 94,(WORK(16,L),L=2,20,2)
 94 FORMAT(4X,10F12.1)
901 CONTINUE
    CALL EXIT
    END
```

```
      SUBROUTINE NDIST(Y,XMU,SIGMA)
      D=0.0
      DO 490 J=1,12
      S=RANF(-1)
  490 D=D+S
      Y=(D-6.)*SIGMA+XMU
      RETURN
      END
      SUBROUTINE UNDIST(Y,A,B)
      Y=RANF(-1)
      Y=Y*(B-A)+A
      RETURN
      END
```

```
      SUBROUTINE EXPEXT(FX,X)
      R=RANF(-1)
      X=-FX*LOGF(R)
      RETURN
      END
```

Operating Instructions (Time Sharing Version)

The Input Data for ARF-1 are:

The following data must occur in the order specified.

A. The run title. 32 or fewer characters, alphanumeric.

B. Number of students and faculty for work occurrence calculations. Include a decimal point, and separate the two numbers by",".

C. The number of staff descriptor lines which follow. 15 or fewer, no decimal point.

D. That number of staff descriptor lines. Each line has the following elements, whose correct placement is critical:

1) line number - 1, 2, or 3 digits usually.
2) 1 space
3) 2-digit index to staff descriptor, followed by 1 space.
4) 2-digit indicator of the number of people of this type, followed by 1 space.
5) 5-digit salary for 1 person of this type, followed by 1 space.
6) 2-digit normal hours in the work week for this person, followed by 1 space.
7) 2-digit overtime hours in the work week available for this person, if needed, followed by 1 space.
8) 32 (or fewer) characters of descriptive information.

There are no decimal points in this data. Leading zeros are to be included for spacing purposes: e.g. 3 would be keyed as 03.

E. The number of work descriptor lines which follow. 15 or fewer, no decimal point.

F. That number of work descriptor lines. Each line has the following elements, whose correct placement is critical:

1) line number - as many digits as needed, but usually 1, 2, or 3.

2) 1 space

3) 2-digit index to work type descriptor, followed by 1 space.
No decimal point.

4) 2-digit index to type of distribution to be sampled, followed by
1 space. 01 is uniform distribution;
02 is normal distribution, and
03 is exponential decay distribution.
No decimal point.

5) Minimum time for one occurrence of this type of work, in tenths of a time unit (hours), keyed as digit-decimal point-digit-space.

6) 3-position maximum time, as in 5 above, followed by 1 space.

7) Multiplier for number of students to determine contribution of students to workload. In essence, this number is the number of times per week each student will require this kind of work. Format is digit-decimal point-2 digits-space.

8) Multiplier for number of faculty. Four digits, decimal point, space as in 7 above.

9) Constant to be added. Same format as in 7 above.

10) Unit of change for this type of work, in same format as 7 above. This is really the fraction represented by 1 unit of change in the weekly profile. If the profile is 1.0 and the unit of change is 0.50, there will be 50% more occurrences of this type of work than would be predicted from the number of students, faculty, and the given constant.

11) 32 or fewer descriptive characters to be printed whenever this type of work is referred to in the output.

G. The number of staff work descriptor lines which follow (15 or fewer), each of which may be split between 2 computer lines, the number of work descriptions on the first line (9 or fewer), and the number of work descriptions on the second line (3 or fewer). 0 must be keyed if the second descriptor line does not occur. All staff types must have the same number of descriptor lines. 3 numbers, separated by commas, no decimal points.

H. The number of descriptor lines, or pairs of descriptor lines, indicated in G above. Each line has the following elements, whose correct placement is critical:

1) line number - as many digits as necessary.
2) 1 space
3) 2-digit staff type indicator, followed by 1 space.
4) As many work descriptions as indicated in the second number in G above. Each work description is:
    a) 2-digit work type indicator, followed by 1 space.
    b) 2-digit minimum or average percent of this type of staff person's time to be spent on this type of work. 1 blank space.
    c) 2-digit maximum percent (or 00 if b above is to be interpreted as average time) of time to be spent on this work, 1 space.

The second line, if used, is in the same format. It will have a higher line number, the same staff type indicator, and the number of work descriptions called for by the 3rd number in G above.

No decimal points occur in this sub-file.

ARF (Adult Resources Flow) Operating Instructions - time sharing

1. Fetch the program.	FETCH ARF1
2. Set the random number generator. You may elect to do nothing, in which case every run will use the same sequence of "random" numbers. Or, you may put in a new seed and get a different sequence. You select a 9 or 10 digit number to put in as shown.	NAME ARF1 16 ISEED - 9574639672
3. Either fetch the data to be used, or create a named string containing the data files needed. Check and edit your data as necessary before running. You may change any line of data by merely keying another line with the same line number.	FETCH ARFDATA or NAME DATA 1 TEST RUN FOR. . . etc.
4. If necessary, reorder your strings so that the data string immediately follows the program. This will not normally be necessary if you fetched program and fetched or created data in order, but will be necessary if you want to run the program with more than 1 set of data.	
5. Give the run command.	RUN ARF1

6. The computer will ask if you want a new week. Type YES if you do, NO to end the run. You may run as many weeks as you like by simply answering YES each time the question is asked.

```
DO YOU WANT A NEW WEEK?
```

7. The computer will ask if you want a new profile. You answer YES or NO. If NO, the existing profile will be used (and, initially, that will be all zero).

```
DO YOU WANT A NEW PROFILE?
?NO
```

8. If you answered YES in the previous step, key in the change factors, 1 at a time, as the computer asks for them. 15 will be asked for, even though some work types are not in use.

```
1
?0.0
2
?1.0
   - etc.
```

9. The run will stop when you answer NO to the question in step 6 above. You may also stop the program with CNTRL C, and may suppress printing with ESC (ALT).

Figure 6

```
NAMES

1  ARF1        759
2  ARFDATA     382

RUN ARF1

 8K

DATA FOR STSFF WORK, FLNT, 8-1-69
STUDENTS=700., FACULTY=35.

STAFF TYPE                              NUMBER   COST   TOTAL   HOURS   OVER

 1  TEACHER STAGE 2-4 LEVEL 2               2  14900.  29800.    40.      0
 2  TEACHER STAGE 1 LEVEL 2                 1  14900.  14900.    40.      0
 3  TEACHER COORDINATING LEVEL 3            6  12500.  75000.    40.      0
 4  TEACHER NURSING LEVEL 3                 1  12500.  12500.    40.      0
 5  TEACHER RESOURCE CENTER LEVEL 3         1  12500.  12500.    40.      0
 6  TEACHER GRAPHIC ARTS LEVEL 3            1  12500.  12500.    40.      0
 7  TEACHER SHOP LEVEL 3                    1  12500.  12500.    40.      0
 8  TEACHER AV LEVEL 4                      1   8850.   8850.    40.      0
 9  TEACHER MUSIC LEVEL 4                   1   8850.   8850.    40.      0
10  TEACHER PSYS ED LEVEL 4                 2   8850.  17700.    40.      0
11  TEACHER LEVEL 4                         6   8850.  53100.    40.      0
12  TEACHER-AIDE                           12   5700.  68400.    40.      0
13  STUDENT HELPER - TUTOR                 50    300.  15000.    10.      0

  TOTALS                                   85         341600.  1900.      0

HOURS AVAILABLE FOR TYPES OF WORK, STAFF WORK, FLNT, 8-1-69

TYPE OF WORK                            REGHRS   OVHRS EXREGH    EXOVHR MAXHRS

 1  COUNSELING, PRESCRIBING              184.0       0      0         0  184.0
 2  PLANNING                              74.0       0  108.0         0  182.0
 3  TUTORING                             372.2       0      0         0  372.2
 4  GROUP INSTRUCTION                    188.8       0      0         0  188.8
 8  PARENTS AND COMMUNITY                 11.6       0   23.2         0   34.8
 9  IN SERVICE TRAINING                   70.0       0      0         0   70.0
10  TESTING                              152.2       0   26.6         0  178.8
11  SCORING / RECORDING                  111.0       0   10.0         0  121.0
12  MATERIALS PREPARATION                 42.0       0   84.4         0  126.4
13  SUPPORTING DUTIES                     86.0       0   98.0         0  184.0
14  DISCIPLINE                               0       0   18.0         0   18.0
15  SUPERVISING ACADEMIC ACTIVITY        133.0       0   57.0         0  190.0

  TOTALS                                1424.8       0  425.2         0 1850.0
```

Figure 7

```
WORK DESCRIPTORS FOR STAFF WORK, FLNT, 8-1-69

WORK TYPE                               DIST    RANGE    STU    FAC    CON    CHG

 1  COUNSELING, PRESCRIBING               3    .1- .2   2.75      0      0      0
 2  PLANNING                              3    .5-1.0      0   5.00      0    .50
 3  TUTORING                              3    .1- .3   2.44      0      0    .50
 4  GROUP INSTRUCTION                     1    .4- .8    .10      0      0    .50
 8  PARENTS AND COMMUNITY                 3    .1- .3    .10   1.00      0   3.00
 9  IN SERVICE TRAINING                   1    .4-1.0      0   5.00      0      0
10  TESTING                               3    .4- .8    .40    .50      0    .30
11  SCORING / RECORDING                   2     0- .1   1.00      0      0    .30
12  MATERIALS PREPARATION                 3    .5-1.0      0   1.00      0   1.00
13  SUPPORTING DUTIES                     3    .5-1.5      0   5.00      0    .50
14  DISCIPLINE                            1    .1- .1    .20   4.00      0      0
15  SUPERVISING ACADEMIC ACTIVITY         1    .8-1.5      0   3.00      0    .50

DO YOU WANT A NEW WEEK?
?YES

DO YOU WANT A NEW PROFILE?
?NO

DEMAND FOR WORK IN WEEK 1  STAFF WORK, FLNT, 8-1-69

WORK TYPE                             OCCURS     HOURS

 1  COUNSELING, PRESCRIBING             1925      224.0
 2  PLANNING                             175       96.8
 3  TUTORING                            1708      234.1
 4  GROUP INSTRUCTION                     70       42.4
 8  PARENTS AND COMMUNITY                105       15.7
 9  IN SERVICE TRAINING                  175      122.2
10  TESTING                              297      143.0
11  SCORING / RECORDING                  700       37.6
12  MATERIALS PREPARATION                 35       20.7
13  SUPPORTING DUTIES                    175      122.4
14  DISCIPLINE                           280       28.0
15  SUPERVISING ACADEMIC ACTIVITY        105      120.9

  TOTAL                                          1207.7

DO YOU WANT A NEW WEEK?
?YES

DO YOU WANT A NEW PROFILE?
?NO
```

Figure 8

```
DEMAND FOR WORK IN WEEK 2  STAFF WORK, FLNT, 8-1-69

WORK TYPE                             OCCURS      HOURS

 1  COUNSELING, PRESCRIBING            1925       229.6
 2  PLANNING                            175        96.3
 3  TUTORING                           1708       229.4
 4  GROUP INSTRUCTION                    70        42.3
 8  PARENTS AND COMMUNITY               105        15.2
 9  IN SERVICE TRAINING                 175       118.0
10  TESTING                             297       127.0
11  SCORING / RECORDING                 700        31.0
12  MATERIALS PREPARATION                35        21.2
13  SUPPORTING DUTIES                   175       118.6
14  DISCIPLINE                          280        28.0
15  SUPERVISING ACADEMIC ACTIVITY       105       119.8

  TOTAL                                          1176.5

DO YOU WANT A NEW WEEK?
?YES

DO YOU WANT A NEW PROFILE?
?NO

DEMAND FOR WORK IN WEEK 3  STAFF WORK, FLNT, 8-1-69

WORK TYPE                             OCCURS      HOURS

 1  COUNSELING, PRESCRIBING            1925       237.7
 2  PLANNING                            175        94.5
 3  TUTORING                           1708       230.9
 4  GROUP INSTRUCTION                    70        40.1
 8  PARENTS AND COMMUNITY               105        13.7
 9  IN SERVICE TRAINING                 175       119.5
10  TESTING                             297       144.2
11  SCORING / RECORDING                 700        39.2
12  MATERIALS PREPARATION                35        23.4
13  SUPPORTING DUTIES                   175       127.5
14  DISCIPLINE                          280        28.0
15  SUPERVISING ACADEMIC ACTIVITY       105       120.5

  TOTAL                                          1219.2
```

Figure 9

```
DO YOU WANT A NEW WEEK?
?YES

DO YOU WANT A NEW PROFILE?
?NO

DEMAND FOR WORK IN WEEK 4  STAFF WORK, FLNT, 8-1-69

WORK TYPE                                 OCCURS     HOURS

 1  COUNSELING, PRESCRIBING                1925      231.0
 2  PLANNING                                175      108.1
 3  TUTORING                               1708      230.7
 4  GROUP INSTRUCTION                        70       41.5
 8  PARENTS AND COMMUNITY                   105       14.8
 9  IN SERVICE TRAINING                     175      119.4
10  TESTING                                 297      124.3
11  SCORING / RECORDING                     700       33.9
12  MATERIALS PREPARATION                    35       26.2
13  SUPPORTING DUTIES                       175      132.2
14  DISCIPLINE                              280       28.0
15  SUPERVISING ACADEMIC ACTIVITY           105      122.0

  TOTAL                                             1212.1

DO YOU WANT A NEW WEEK?
?NO

TIME:  2.606 SEC.
```

T.S. ARF

atement No.	Variables	
	STU	# of Students
	FAC	# of Faculty
	IT	Title of Run
55	IN	# of Staff Types (# Descriptors)
	THRS	Total Hours Available (Staff)
	TOVHR	Total Overtime Hours Available (Staff)
	TC	Total Staff Cost
	NT	Total Number of Staff Members
74	K	Staff Type Index Number
	N	Number of Staff Members of that type
	C	Cost of 1 Staff Member of that Type
	HRS(k)	# Hours/Week that type of Staff Member Works
	OVHR(k)	# Overtime Hours/Week that type of Staff Member Works
	IFT	Descriptive Title for that type of Staff Member
100	IN	# of Work Type Descriptors
106	K	Work Type Index Number
	IDTY	Time Distribution Type for that Work Descriptor (Uniform, Normal, Exponential Decay)
	PA	Minimum Time Per Contact for that type of Work (Nearest 1/10 hour)
	PB	Maximum Time Per Contact for that type of Work (Nearest 1/10 hour)
	ST	Number of Times Per Student for Weeks for that type of work
	FA	Number of Times Per Teacher/Faculty for Weeks for that type of work
	WLD	Constant Demand for that type of Work Per Week (Note: Total Demand=ST*STU + FA*FAC + WLD)
	PCT	Fraction of Increase or Decrease in an Unusual Week
	IWT	Descriptive Title for that type of work
115	IN	# of Staff Work Descriptors (Job Descriptions)
	NI	# of Work Types on 1st Line of Description
	N2	# of Work Types on 2nd Line of Description
	N3	Index of 1st Work Type on 2nd Line of Description
	N4	Total # of Work Types/Staff Work Description
124	K	Staff Type Index Number
	ITYP	Work Type Index Number
	TMIN	Minimum of Average % of Time Spent By that Staff Type on That Work Type
	TMAX	Maximum % of Time (If Zero, TMIN is Average, Not Minimum)
	AVL	Hours Available for that type of Work Table
	TQT1	Total Regular Hours Available, Using Min. or Aver. Time
	TWT2	Total Overtime Hours Available, Using Min. or Aver. Time
	TQT3	Additional Reg. Hours Available, Using Max. Time
	TQT4	Additional Overtime Hours Available, Using Max. Time
	GTQT	Grand Total of Hours Available
162	G	Total of All Hours Available for a Work Type
898	N	Week Counter
0,905	M	Console Response (Yes or No)
	TTQT	Total Hours Required for Work that Week

	PFL	Factor to Change Profile of Work Demand By (Work Needed = PCT*PFL*(Total Demand)
	NQC	Number of Occurrences
, 965, 975	EX	Mean of Total Time Distribution (Depends on Distribution Type)
	VARX	Variance of Mean of Total Time Distribution
	SIGMA	Std. Deviation of Mean of Total Time Distribution
	Y	Sample from Normal Distribution WithEX & SIGMA as Parameters

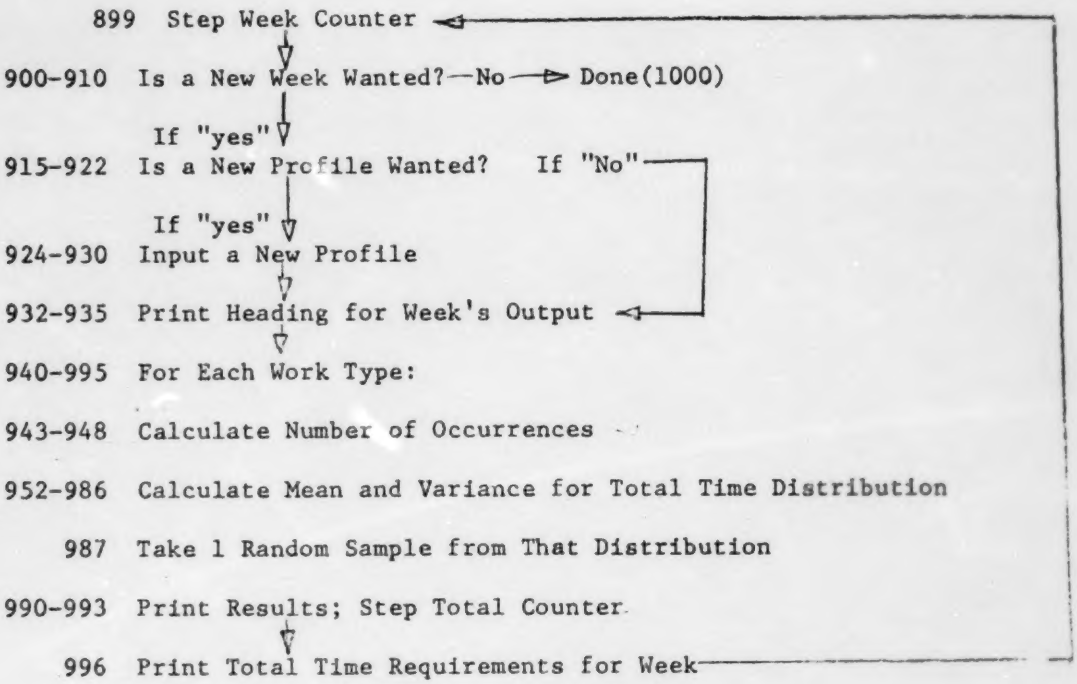
T.S. ARF

Main Routine Logic: Initialization, Reading In and Displaying Data

16-17	Set Random Number Generator Seed
20-35	Set All Tables To 0.0 Initially
40	Read Title For Run
45	Read Number of Students, Faculty For This Run
50	Print Title, Number of Students, Faculty
55	Read in Number of Staff Descriptors
61	Print Heading For Displaying Staff Descriptions
64-70	Initialize Total Counters to Zero
72-90	Read In and Print Staff Descriptors; Total Hours, Costs, # Staff
93	Print Totals for Staff Descriptors
100	Read in Number of Work Descriptors
104-106	Read in Work Descriptors
115	Read in Number of Staff-Work (Job) Descriptors, Line Info
120-143	Read in Staff-Work Descriptors and Calculate Hours Available
150-178	Print Hours Available for Each Type of Work and Totals
200-214	Print Work Descriptors→To Week Simulator Coop (899)

T.S. ARF

Week Simulator Loop Logic



899 Step Week Counter

900-910 Is a New Week Wanted? —No→ Done(1000)

If "yes"

915-922 Is a New Profile Wanted? If "No"

If "yes"

924-930 Input a New Profile

932-935 Print Heading for Week's Output

940-995 For Each Work Type:

943-948 Calculate Number of Occurrences

952-986 Calculate Mean and Variance for Total Time Distribution

987 Take 1 Random Sample from That Distribution

990-993 Print Results; Step Total Counter

996 Print Total Time Requirements for Week

T.S. ARF

Logic of Subroutine NDist

Y = Random Normal Variate from XM, Sigma
XM = Mean of Distribution
SIGMA = Standard Deviation of Distribution
D = Accumulator for 12 Random Uniform (0.1) Numbers
S = sample from Uniform (0,1)

1025 Set D to Zero

1030-1040 Get 12 Uniform Random (0,1) Numbers and Sum in D

1045 Get Y with Desired Mean and Std. Deviation
Using Fact That

Note: Distribution of MEANS of almost any distribution is NORMAL about original distribution mean. Thus for total times it is acceptable to sample a normal distribution regardless of shape of original distribution.

Figure 10   INPUT IS FORMATTED, <u>NOT</u> FREEFIELD.

```
FETCH ARFDATA
OK.  DATE FILED:  10/06/70.

LIST
1  STAFF WORK, FLNT, 8-1-69
2  700.0, 35.0
3  13
4  01  02  14900  40  00  TEACHER STAGE 2-4 LEVEL 2
5  02  01  14900  40  00  TEACHER STAGE 1 LEVEL 2
6  03  06  12500  40  00  TEACHER COORDINATING LEVEL 3
7  04  01  12500  40  00  TEACHER NURSING LEVEL 3
8  05  01  12500  40  00  TEACHER RESOURCE CENTER LEVEL 3
9  06  01  12500  40  00  TEACHER GRAPHIC ARTS LEVEL 3
10  07  01  12500  40  00  TEACHER SHOP LEVEL 3
11  08  01  08850  40  00  TEACHER AV LEVEL 4
12  09  01  08850  40  00  TEACHER MUSIC LEVEL 4
13  10  02  08850  40  00  TEACHER PHYS ED LEVEL 4
14  11  06  08850  40  00  TEACHER LEVEL 4
15  12  12  05700  40  00  TEACHER-AIDE
16  13  50  00300  10  00  STUDENT HELPER - TUTOR
20  12
21  01  03  0.1  0.2  2.75  0.00  0.00  0.00  COUNSELING, PRESCRIBING
22  02  03  0.5  1.0  0.00  5.00  0.00  0.50  PLANNING
23  03  03  0.1  0.3  2.44  0.00  0.00  0.50  TUTORING
24  04  01  0.4  0.8  0.10  0.00  0.00  0.50  GROUP INSTRUCTION
25  08  03  0.1  0.3  0.10  1.00  0.00  3.00  PARENTS AND COMMUNITY
26  09  01  0.4  1.0  0.00  5.00  0.00  0.00  IN SERVICE TRAINING
27  10  03  0.4  0.8  0.40  0.50  0.00  0.30  TESTING
28  11  02  0.0  0.1  1.00  0.00  0.00  0.30  SCORING / RECORDING
29  12  03  0.5  1.0  0.00  1.00  0.00  1.00  MATERIALS PREPARATION
30  13  03  0.5  1.5  0.00  5.00  0.00  0.50  SUPPORTING DUTIES
31  14  01  0.1  0.1  0.20  4.00  0.00  0.00  DISCIPLINE
32  15  01  0.8  1.5  0.00  3.00  0.00  0.50  SUPERVISING ACADEMIC ACTIVITY
50  13,6,3
51  01  01  20  00  02  10  20  03  25  00  04  20  00  08  01  03  09  05  00
52  01  12  00  06  14  00  02  15  07  10
53  02  01  20  00  02  10  20  03  25  00  04  20  00  08  01  03  09  05  00
54  02  12  00  06  14  00  02  15  07  10
55  03  01  20  00  02  10  20  03  25  00  04  20  00  08  01  03  09  05  00
56  03  12  00  06  14  00  02  15  07  10
57  04  01  20  00  02  05  10  03  22  00  04  20  00  08  01  03  09  05  00
58  04  12  00  06  13  10  00  15  07  10
59  05  01  20  00  02  05  10  03  15  00  04  10  00  08  01  03  09  05  00
60  05  10  14  00  12  20  40  15  07  10
61  06  01  20  00  02  05  10  03  09  00  04  13  00  08  01  03  09  05  00
62  06  12  05  10  13  10  00  15  07  10
63  07  01  20  00  02  05  10  03  22  00  04  20  00  08  01  03  09  05  00
64  07  12  05  10  13  05  10  15  07  10
65  08  01  20  00  02  05  10  03  10  00  04  20  00  08  01  03  09  05  00
66  08  12  10  20  13  10  00  15  07  10
67  09  01  20  00  02  05  10  03  14  00  04  33  00  08  01  03  09  05  00
68  09  12  05  10  14  00  02  15  07  10
69  10  01  20  00  02  05  10  03  09  00  04  40  00  08  01  03  09  05  00
70  10  10  02  04  12  00  05  15  07  10
71  11  01  20  00  02  05  10  03  24  00  04  16  00  09  05  00  10  10  00
72  11  12  00  06  13  10  20  15  07  10
73  12  02  00  05  03  22  00  08  01  03  09  05  00  10  20  00  11  20  00
74  12  12  05  10  13  10  25  15  07  10
75  13  02  02  06  03  15  00  10  05  10  11  03  05  14  00  02  15  07  10
76  13  00  00  00  00  00  00  00  00  00
```

```
5 PROGRAM ARF
10 DIMENSION AVL (15,4),IT(4),IWT(15,4),IFT(4),HRS(15),OVHR(15)
11 DIMENSION PFL(15)
12 DIMENSION ITYP(12),TMIN(12),TXAX(12)
15 DIMENSION IDTY(15),PA(15),PB(15),ST(15),FA(15),WLD(15),PCT(15)
16 ISEED=74938731957
17 CALL RANFSET(ISEED)
20 DO 35  I=1,15
21 HRS(I)=0.0
22 OVHR(I)=0.0
23 IDTY(I)=0
24 PA(I)=0.0
25 PB(I)=0.0
26 ST(I)=0.0
27 FA(I)=0.0
28 WLD(I)=0.0
29 PCT(I)=00
              30 PFL(I)=0.0
33 DO 35  J=1,4
35 AVL(I,J)=0.0
40 READ 41, IT
41 FORMAT (S,4A8)
45 READ, STU, FAC
50 PRINT 51, IT, STU, FAC
51 FORMAT (//,*DATA FOR *,4A8,/,*STUDENTS=*,F4.0,*, FACULTY=*,
52C F3.0)
55 READ, IN
60 IF (IN.LE.0) GO TO 100
61 PRINT 62
62 FORMAT (//,*STAFF TYPE*,26X,*NUMBER*,3X,*COST   TOTAL*,
63C 3X,*HOURS OVER*,/)
64 THRS=0.0
66 TOVHR=0.0
68 TC=0.0
70 NT=0
72 DO 90  I=1,IN
74 READ 75,K,N,C,HRS(K),OVHR(K),IFT
75 FORMAT (S,I3,I3,F6.0,2F3.0,1X,4A8)
76 X=N
77 THRS=THRS+HRS(I)*X
78 TOVHR=TOVHR+OVHR(I)*X
80 CT=C*X
81 TC=TC+CT
82 NT=NT+N
86 PRINT 87, K, IFT,N,C,CT,HRS(K),OVHR(K)
87 FORMAT (1X,I2,2X,4A8,I5,2F8.0,F8.0,F5.0)
88 HRS(K)=HRS(K)*X
89 OVHR(K)=OVHR(K)*X
90 CONTINUE
93 PRINT 94,NT,TC,THRS,TOVHR
94 FORMAT (/,3X,*TOTALS*,28X,I5,8X,2F8.0,F5.0)
100 READ, IN
102  IF (IN .LE. 0)  GO TO 1000
```

```
104 DO 136  I=1,IN
106 READ 108,K,IDTY(K),PA(K),PB(K),ST(K),FA(K),WLD(K),PCT(K),
107C (IWT(K,J),J=1,4)
108 FORMAT (S,2I3,2F4.1,4F5.2,1X,4A8)
115 READ, IN,N1,N2
116 N3=N1+1
117 N4=N1+N2
118 IF (IN.LE.0) GO TO 898
120 DO 143  I=1,IN
124 READ 125,K,(ITYP(J),TMIN(J),TXAX(J),J=1,N1)
125 FORMAT (S,I3,6(I3,2F3.0))
126 IF (N2 .LE. 0) GO TO 131
128 READ 125,K,(ITYP(J),TMIN(J),TXAX(J),J=N3,N4)
131 DO 142  M=1,N4
133 L=ITYP(M)
135 IF (L .LE. 0) GO TO 142
137 AVL(L,1)=AVL(L,1)+TMIN(M)*HRS(K)/100.0
138 AVL(L,2)=AVL(L,2)+TMIN(M)*OVHR(K)/100.0
139 IF (TXAX(M).LE.0.0) GO TO 142
140 AVL(L,3)=AVL(L,3)+(TXAX(M)-TMIN(M))*HRS(K)/100.0
141 AVL(L,4)=AVL(L,4)+(TXAX(M)-TMIN(X))*OVHR(K)/100.0
142 CONTINUE
143 CONTINUE
150 PRINT 151, IT
151 FORMAT (//,*HOURS AVAILABLE FOR TYPES OF WORK, *,4A8)
152 PRINT 153
153 FORMAT(/,*TYPE OF WORK*,25X,*REGHRS  OVHRS EXREGH EXOVHR*,
154C * MAXHRS*)
155 TOT1=0.0
156 TOT2=0.0
157 TOT3=0.0
158 TOT4=0.0
159 GTOT=0.0
160 DO 175  I=1,15
162 G=AVL(I,1)+AVL(I,2)+AVL(I,3)+AVL(I,4)
163 GTOT=GTOT+G
164 IF (G.GT.0.0) GO TO 166
165 IF (IDTY(I).EQ.0) GO TO 175
166 TOT1=TOT1+AVL(I,1)
168 TOT2=TOT2+AVL(I,2)
169 TOT3=TOT3+AVL(I,3)
170 TOT4=TOT4+AVL(I,4)
172 PRINT 173,I,(IWT(I,J),J=1,4),(AVL(I,J),J=1,4),G
173 FORMAT (1X,I2,2X,4A8,5F7.1)
175 CONTINUE
177 PRINT 178,TOT1,TOT2,TOT3,TOT4,GTOT
178 FORMAT (/,3X,*TOTALS*(28X,5F7.1)
200 PRINT 201, IT
201 FORMAT (///,*WORK DESCRIPTORS FOR*,4A8)
2003 PRINT 204
204 FORMAT (/,*WORK TYPE*,26X,*DIST   RANGE*,4X,*STU  FAC  CON*,
205C *  CHG*,/)
207 DO 214  I=1,15
209 IF (IDTY(I).EQ.0) GO TO 214
211 PRINT 213,I,(IWT(I,J),J=1,4),IDTY(I),PA(I),PB(I),ST(I),FA(I),
212C WLD(I),PCT(I)
213 FORMAT (1X,I2,2X,4A8,I2,2X,F3.1,1H-,F3.1,2X,4F5.2)
214 CONTINUE
898 N=0
899 N=N+1
900 PRINT 901
901 FORMAT (/,*DO YOU WANT A NEW WEEK?*)
905 INPUT, M
910 IF (M .NE. 8HYES      ) GO TO 1000
912 TTOT=0.0
915 PRINT 916
916 FORMAT (/,*DO YOU WANT A NEW PROFILE?*)
920 INPUT, M
922 IF (M .NE. 8HYES      ) GO TO 932
924 PRINT 925
925 FORMAT (/,*TYPE IN CHANGE FACTORS, F FORMAT, 1 AT A TIME*)
927 DO 930  I=1,15
928 PRINT 929,I
929 FORMAT (1X,I2)
930 INPUT, PFL(I)
932 PRINT 933,N,IT
933 FORMAT (/,*DEMAND FOR WORK IN WEEK *,I2,2X,4A8)
934 PRINT 935
935 FORMAT (/,*WORK TYPE*,27X,*OCCURS*,3X,*HOURS*,/)
940 DO 995 I=1,15
941 IF (IDTY(I).LE.0) GO TO 995
942 IF (IDTY(I).GT.3) GO TO 995
943 M=IDTY(I)
945 X=ST(I)*STU+FA(I)*FAC+WLD(I)
948 NOC=X*(1.0+PFL(I)*PCT(I))
950 IF (NOC .LE. 0)  GO TO 995
952 GO TO (955,965,975), M
955 EX=(PA(I)+PB(I))/2.0
960 VARX=((PB(I)-PA(I))**2)/12.0
962 GO TO 980
965 EX=(PA(I)+PB(I))/2.0
970 VARX=(PB(I)-PA(I))/6.0
972 GO TO 980
975 EX=(PB(I)-PA(I))*0.20+PA(I)
978 VARX=EX**2
980 Y=NOC
982 EX=Y*EX
984 VARX=Y*VARX
986  SIGMA=SQRT(VARX)
988 CALL NDIST (Y,EX,SIGMA)
```

```
990 TTOT=TTOT+Y
992 PRINT 994,I,(IWT(I,J),J=1,4),NOC,Y
994 FORMAT (1X,I2,2X,4A8,I5,F9.1)
995 CONTINUE
996 PRINT 997,TTOT
997 FORMAT (/,3X,*TOTAL*,34X,F9.1)
998 GO TO 899
1000 END
1020 SUBROUTINE NDIST (Y,XM,SIGMA)
1025 D=0.0
1030 DO 1040  J=1,12
1035 S=RANF(-1)
1040 D=D+S
1045 Y=(D-6.0)*SIGMA + XM
1050 RETURN
1055 END
1060 ENDPROG
```

EDSIM I

The EDSIM I[1] model is the first attempt to follow the progress of individual students through a new, individualized program. The relatively simple level of operation is very appropriate for the early stages of program planning, when detailed data is not available, and when initial concepts and assumptions are readily modified.

The major variable considered is time. The model traces a specified number of students through a specified number of randomly-selected instructional events in each of several pedagogical areas. From input data about each event, a tabulation is made of the following:

a. Student time to completion of program, in event-hours. Real time, of course, would be quite different, depending on ability to schedule events a student needs and on his willingness to put in varying numbers of hours per day.

b. Student time in each pedagogical area, in event-hours.

c. Relative demands for different types of events in different pedagogical areas.

The operation of EDSIM I is best explained through examination of the output it produces.

---

[1] This model was developed as part of the Model Elementary Teacher Education Project at the University of Massachusetts. Reports of the Project are available through ERIC as follows:
"A Feasibility Study on the METEP", Phase II, Vol 1. Ed 043 - 582.
"A Feasibility Study on the METEP", Phase II, Vol. II. Final Report. Ed 043 - 583.
"Summary: A Feasibility Study on the METEP", Phase II. Final Report. Ed 043 - 584.

Computer Output number 1 displays the following:

a. Run identification: title and number. (It becomes very easy to confuse the outputs from a number of different runs.)

b. Number of students in each profile (input data).

c. Number of profiles in this run (input data).

d. The probabilities of passing pre-tests and post-tests in each pedagogical area (input data).

A profile such as that shown in part for the Human Relations Specialist in the next output, is basically the number of performance criteria to be met in each pedagogical area. Specific performance criteria, and specific instructional alternatives to meet them, had not yet been designed when this model was created; hence the use of the number of criteria, and the distribution of instructional events.

Computer Output number 2 shows the following:

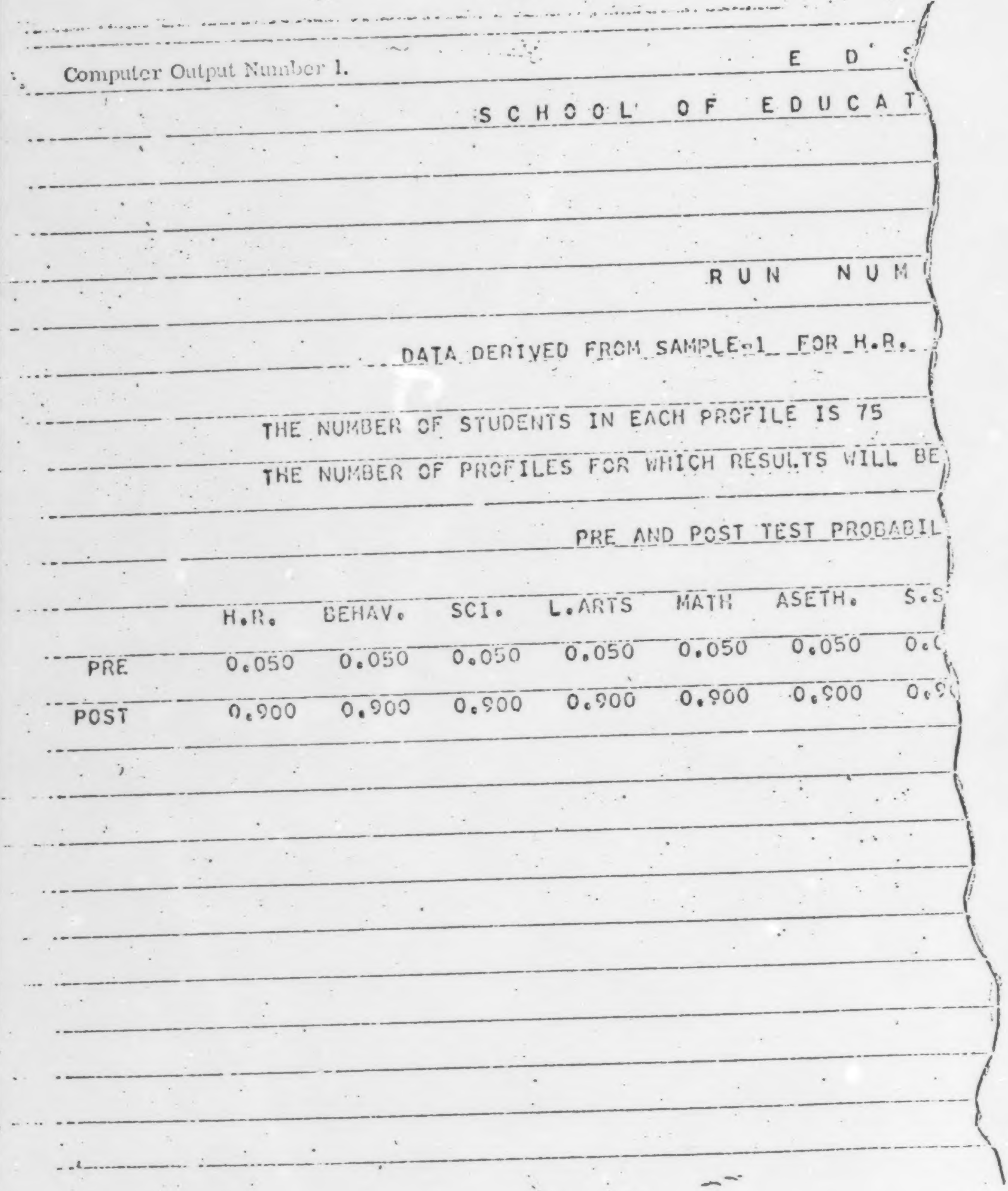
a. Indication of the specific profile for which the following data applies.

b. The event type selection probabilities in each pedagogical area. These need not be the same in each area, but happen to be for the run shown because of lack, at the time, of any better data to put in.

c. The fraction of students passing various numbers of performance criteria by means of successfully taking a pretest. This is created by sampling the given probability for passing a pre-test the number of times called for in the specific profile. Since this is a random sampling we expect fluctuation from run to run. One purpose of this model is to help give educators an idea of what variability might be expected.

d. The fraction of students failing various numbers of post-tests, and, hence, needing additional instruction. This is based on a random sampling of the post-test probability distribution given. We expect, incidentally, that the more

Computer Output Number 1.



```
                                                   E D S
                              S C H O O L   O F   E D U C A T

                                                   R U N   N U M

                              DATA DERIVED FROM SAMPLE-1  FOR H.R.

                THE NUMBER OF STUDENTS IN EACH PROFILE IS 75
                THE NUMBER OF PROFILES FOR WHICH RESULTS WILL BE

                                          PRE AND POST TEST PROBABIL

           H.R.     BEHAV.    SCI.     L.ARTS    MATH     ASETH.    S.S
   PRE     0.050    0.050     0.050    0.050     0.050    0.050     0.
   POST    0.900    0.900     0.900    0.900     0.900    0.900     0.
```

Computer Output Number 2

```
                                                          OUTPUT STATISTICS FOR PROFIL

                                                               HUMAN RELATIONS SPECIAL

                                                    NUMBER OF PERFORMANCE CRITERIA IN

                  H.R.    BEHAV.   SCI.   L.ARTS   MATH   ASETH.   S.S.   FOR.L.

      PC            50      50       20      40      20      20       20      20


                                                           EVENT TYPE SELECTION PROBA

                  H.R.    BEHAV.   SCI.   L.ARTS   MATH   ASETH.   S.S.   FOR.L.

      ET 1       0.250   0.250   0.250   0.250   0.250   0.250   0.250   0.250
      ET 2       0.200   0.200   0.200   0.200   0.200   0.200   0.200   0.200
      ET 3       0.150   0.150   0.150   0.150   0.150   0.150   0.150   0.150
      ET 4       0.100   0.100   0.100   0.100   0.100   0.100   0.100   0.100
      ET 5       0.100   0.100   0.100   0.100   0.100   0.100   0.100   0.100
      ET 6       0.080   0.080   0.080   0.080   0.080   0.080   0.080   0.080
      ET 7       0.050   0.050   0.050   0.050   0.050   0.050   0.050   0.050
      ET 8       0.050   0.050   0.050   0.050   0.050   0.050   0.050   0.050
      ET 9       0.010   0.010   0.010   0.010   0.010   0.010   0.010   0.010
      ET10       0.010   0.010   0.010   0.010   0.010   0.010   0.010   0.010


                                                    PERCENTAGES OF STUDENTS PASSING
   PERCENT
   PASSING
                  H.R.    BEHAV.   SCI.   L.ARTS   MATH   ASETH.   S.S.   FOR.L.

  AT LEAST  1     0.573   0.613   0.293   0.520   0.320   0.320   0.293   0.427
  AT LEAST  2     0.173   0.253   0.093   0.107   0.053   0.093   0.067   0.093
  AT LEAST  3     0.053   0.040   0.000   0.000   0.000   0.013   0.000   0.000
  AT LEAST  4     0.027   0.000   0.000   0.000   0.000   0.000   0.000   0.000
  AT LEAST  5     0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
  AT LEAST  6     0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
  AT LEAST  7     0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
  AT LEAST  8     0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
  AT LEAST  9     0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
  AT LEAST 10     0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000


                                                    PERCENTAGES OF STUDENTS FAILING
   PERCENT
   FAILING
                  H.R.    BEHAV.   SCI.   L.ARTS   MATH   ASETH.   S.S.   FOR.L.

  AT LEAST  1     1.000   1.000   0.960   1.000   0.960   0.987   0.933   0.947
  AT LEAST  2     1.000   0.987   0.813   0.973   0.760   0.813   0.707   0.867
  AT LEAST  3     1.000   0.987   0.440   0.867   0.413   0.533   0.387   0.507
  AT LEAST  4     0.973   0.893   0.280   0.760   0.213   0.227   0.227   0.280
  AT LEAST  5     0.907   0.813   0.133   0.573   0.080   0.067   0.107   0.147
  AT LEAST  6     0.720   0.653   0.080   0.293   0.053   0.013   0.040   0.040
  AT LEAST  7     0.573   0.480   0.027   0.173   0.000   0.000   0.013   0.013
  AT LEAST  8     0.400   0.280   0.000   0.080   0.000   0.000   0.000   0.000
  AT LEAST  9     0.213   0.187   0.000   0.040   0.000   0.000   0.000   0.000
  AT LEAST 10
```

performance criteria there are to meet, the more likely it is that a student will fail occasionally on a random basis.

An event type is specified by its probability of being selected and the length of time in hours required for its completion. Although it may be hypothesized in the real world that completion times, and probably the probability of event type selection, may be a function of student or pedagogical characteristics, no provision has been made in EDSIM I for such variability.

Each student in a given profile starts with a given number of performance criteria to meet in each area. For each area in turn, the pre-test probability distribution is sampled; the number of performance criteria to be met in that area is reduced by the number of pre-test successes.

It is then assumed that each remaining performance criterion to be met will result in a demand for one instructional event. Further, the probability of failing the post-test is sampled the number of times indicated by this number of instructional events to be taken; each failure results in the addition of one more instructional event to the number to be taken.

The number of instructional events, thus adjusted by pre-test successes and post-test failures, is the number of times the event type distribution for the pedagogical area is sampled to give a number of specific event types for the student.

Computer Output number 3 shows the "actual" demand for each event type by area, as "determined" by the process described above.

By summing the hours required for each selected event, the time requirement for each student in each area is "determined".

Computer Output Number 3

```
                                               DEMAND FOR EVENT TYPES-WITH 10

             H.R.   BEHAV.   SCI.   L.ARTS   MATH   ASETH.   S.S.   FOR.

ET 1          985    1021     397     744     375     362     377     40
ET 2          780     780     306     617     300     337     300     37
ET 3          585     556     232     428     224     228     216     2
ET 4          382     388     172     341     175     148     180     1
ET 5          403     373     167     299     152     150     130     13
ET 6          313     287     107     261     108     116     129     12
ET 7          181     200      86     148      81      76      87     75
ET 8          212     212      54     151      78      81      82     74
ET 9           37      34      15      28      20      16      10     18
ET10           37      44      16      35      20      11      14     13

                   SUM OF ABOVE DEMAND - BY TYPE

                             6536    5351    3801    2757    2580    2059
```

Computer Output number 4 indicates:

a. The maximum time for any student in this exercise to complete his work. This is in event-hours, not real time.

b. The minimum time any student completed his work.

c. The average time requirement.

d. A histogram of completion times.

e. The average time for this profile each student spends in each pedagogical area. The example shown, for a Human Relations specialist, indicates an unusual time commitment needed for Mathematics and Science.

Experience with this model in the early development of the Model Elementary Teacher Education Project indicated the following:

1. Present academic programs have relatively little variability, and hence are trivial for purposes of simulation.

2. The probable effects of varying the pre-test and post-test probabilities were examined. For example, changing the probability of passing a pre-test from .05 to .01 and that of a post-test from .90 to .85 will add 10% (or about 200 hours) to the time required to complete a program.

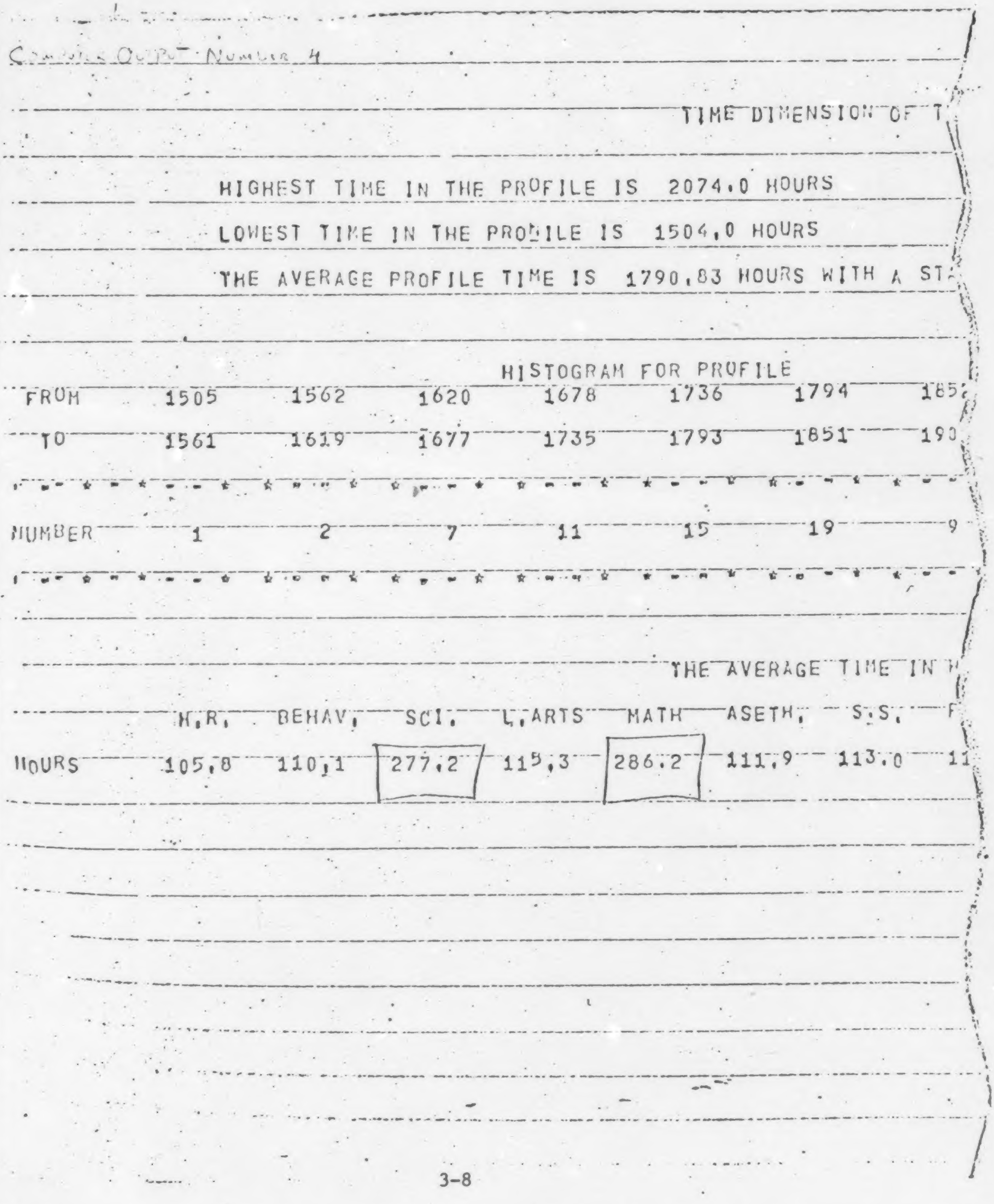
3. The probable effects of changing the total number of performance criteria to be met were examined. For instance, changing the number of required performance criteria from 340 to 400 would add about 20% (or 400 hours) to the completion time.

4. It is difficult for program planners to specify operational details for, and to quantify, the proposed experiences, as indicated by the relatively uniform numbers appearing across areas in the data. Group concensus is not a substitute for detailed planning.

5. In some cases, the perceptions of planners in different pedagogical areas differed markedly, and resulted in a potential program that would not mesh well for students. For instance, the output shown as Computer Output number 4 indicates an unduly long time commitment, at least in relation to other areas, in Mathematics and Science.

The logic of the EDSIM I computer program, and detailed operating instructions, are given on the following pages.

Mathematics and Science areas, as shown on the following partial sample output, has the student spent an unreasonable amount of time in these areas.

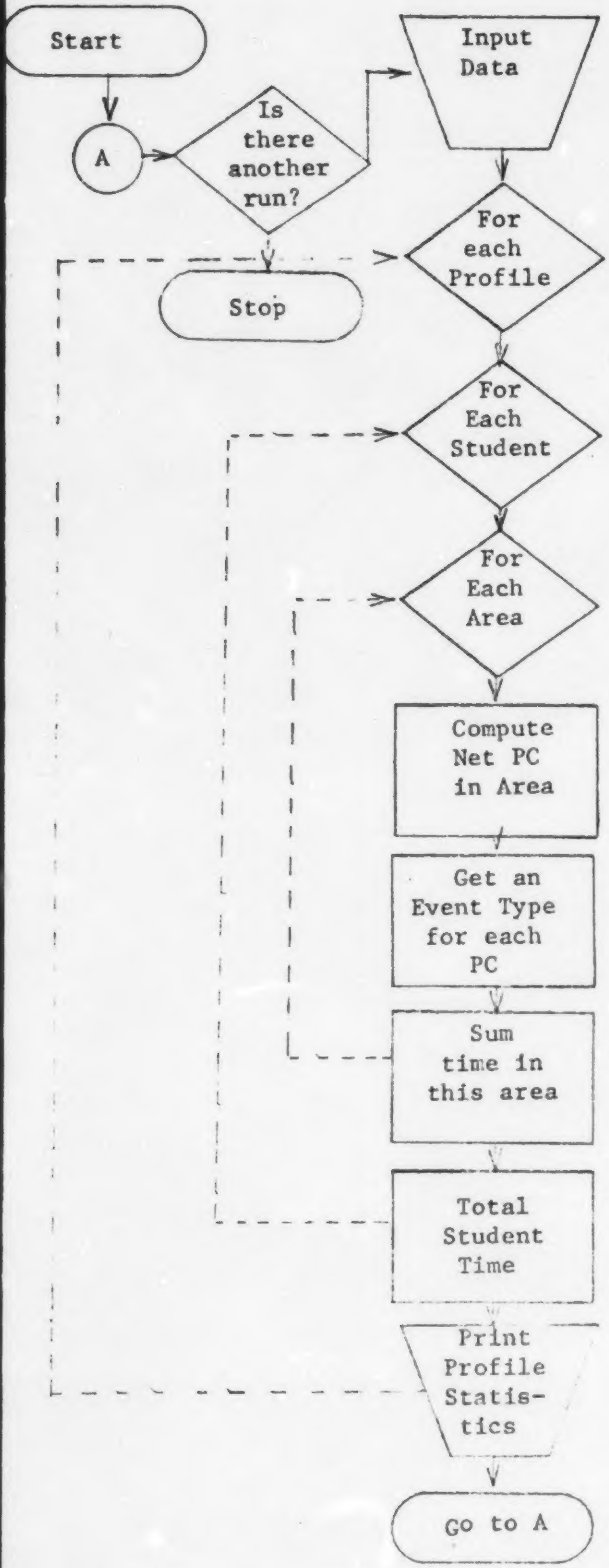


Computer Output Number 4

```
                                                      TIME DIMENSION OF T

                 HIGHEST TIME IN THE PROFILE IS  2074.0 HOURS
                 LOWEST TIME IN THE PROFILE IS  1504.0 HOURS
                 THE AVERAGE PROFILE TIME IS  1790.83 HOURS WITH A ST

                                             HISTOGRAM FOR PROFILE
 FROM      1505      1562      1620      1678      1736      1794      185
  TO       1561      1619      1677      1735      1793      1851      190
 *---*---*---*---*---*---*---*---*---*---*---*---*---*---*---*---*---*---*--
NUMBER        1         2         7        11        15        19         9
 *---*---*---*---*---*---*---*---*---*---*---*---*---*---*---*---*---*---*--

                                                      THE AVERAGE TIME IN H
           H.R.   BEHAV.    SCI.    L.ARTS    MATH    ASETH.    S.S.     F
HOURS     105.8   110.1    277.2    115.3    286.2    111.9    113.0     11
```

FIGURE 1.

BASIC FLOWCHART FOR FIRST STAGE MODEL


Start
A
Is there another run?
Stop
Input Data
For each Profile
For Each Student
For Each Area
Compute Net PC in Area
Get an Event Type for each PC
Sum time in this area
Total Student Time
Print Profile Statis-tics
Go to A

-COMMENTS-

input data for a given run

do the following for each profile in the run

do the following for each student in this profile

do the following for each area in the profile

compute the net number of PC the student must meet in the area

Select an Event Type for each PC

from the number and time of each Event Type, sum the time spent in the area

sum the area times to get total student time in this profile

compute and print summary statistics

see if another run is to be excuted

INPUT DATA FOR EDSIM I

1. Number of different profiles to be tested (13 or fewer)

2. Number of students in each profile (same for all)

3. Number of instructional event types available in each profile (same for all)

4. For each profile, the number of performance criteria to be met in each of up to 13 areas of instruction.

5. For each area of instruction:

   a. The probability of passing a pre-test, and needing no instructional event to satisfy one performance criterion (same for all criteria in an area).

   b. The probability of passing a post-test, and not needing a second instructional event to satisfy one performance criterion (same for all performance criteria in an area).

6. The time necessary to complete each instructional event type (regardless of area in which it occurs).

7. For each area, the distribution of event types (probability of selecting each of the available event types to meet a performance criterion in that area for which a pre-test was not passed).

8. Identifying comments (40 characters) for output.

EDSIM 1 Input Formats, As the Data Deck Should be Submitted

Parameter Card

COLUMN	CONTAINS
1-4	Number of students in run (≤ 9999) I Format
5-8	Number of event types (≤ 25) for before post-test distributions I Format
9-12	Some information for event types for after post-test distributions (usually the same as before post-test distributions but could be different)
15-16	Number of Profiles (≤ 13) I Format
19	L or A for level of detail desired

Profile Card

(one per profile indicated on parameter card)

1-5	Numeration (if operated from detail of level would be 1; if operated from detail of A (area) would be number of skills (objectives) in the unit.
6-10	Place Value
11-15	Addition
16-20	Subtraction
21-25	Multiplication
26-30	Division
31-35	Combination of Processes
36-40	Fractions
41-45	Money
46-50	Time

COLUMN	CONTAINS
51-55	Systems of Measurement
6-60	Geometry
61-65	Special Topics

Pretest Probability Card

(one card required)

1-5	Probability of passing a pretest for numeration
6-10 to 61-65	Some information for other 12 areas listed on Profile Card

Post-Test Probability Card

(one card required)

1-5	Probability of passing a post-test for numeration
6-10 to 61-65	Some information for other 12 areas on Profile Card

Event Type Time Cards

(5 cards are required, even if blank)

		on subsequent card:
1-5	Time in hours to complete event type 1	(6, 11, 16, 21)
6-10	Time in hours to complete event type 2	(7, 12, 17, 22)
11-15	Time in hours to complete event type 3	(8, 13, 18, 23)
16-20	Time in hours to complete event type 4	(9, 14, 19, 24)
21-25	Time in hours to complete event type 5	(10, 15, 20, 25)

All read as F 5.0, but decimal point may be included to give fractional hours.

Event Type Distribution Cards

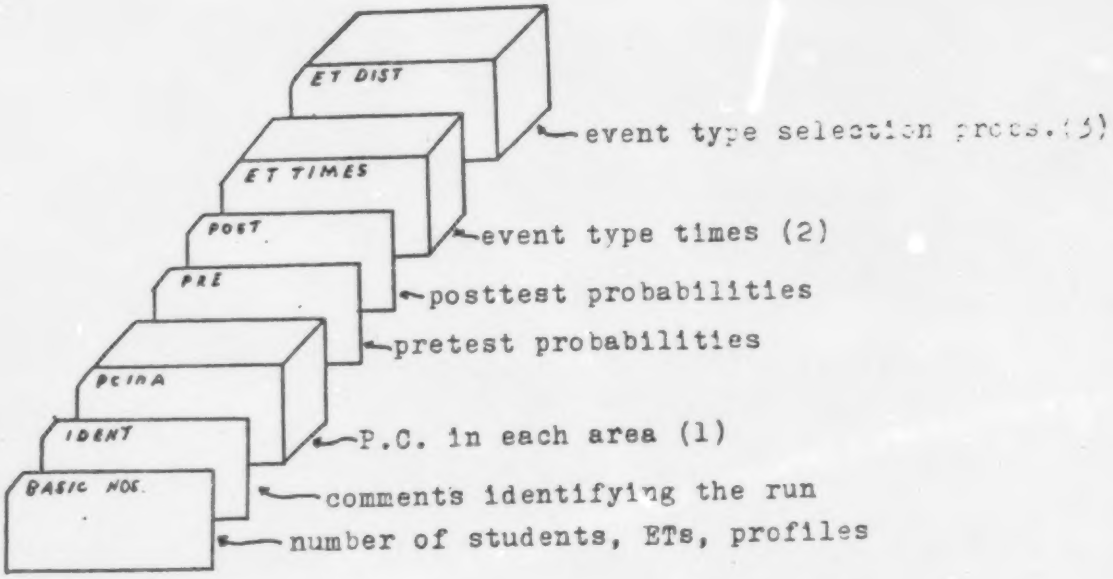
(13 cards required, 1 per area, in order)

COLUMN	CONTAINS
1-3	Selection Probability, event type 1
4-6	Selection Probability, event type 2
73-75	Selection Probability, event type 3 on through Selection Probability, event type 25
	All read as F 3.0; must be keyed as .xx

More than one set of data may be submitted at a time. The program will operate until an end of file condition is encountered.

The FORTRAN formats for each card are:

BASIC NUMBERS	(3I4)
IDENT	(5A8)
PCINA	(13F5.0)
PRE	(13F5.0)
POST	(13F5.0)
E.T. TIMES	(5F5.0)
E.T. DIST	(25F3.0)


ET DIST
event type selection probs.(3)
ET TIMES
event type times (2)
POST
posttest probabilities
PRE
pretest probabilities
PCINA
P.C. in each area (1)
IDENT
comments identifying the run
BASIC NOS.
number of students, ETs, profiles

(1) one card for each profile being considered

(2) five cards total, all must be included-even if blank

(3) 13 cards, one for each area

```
      PROGRAM FDSIM1
      COMMON /A/ PCINA(12.13), APRE(13), APOST(13), TPCINA (13),
     1PASS(13.10), FAIL(13.10), DCUM(13.25), ETSUM(13.25), NOET,
     1SCUM(13.25).NO2ET
      DIMENSION FTIMF(25), ETDIST(13.25).TSUM(13).TMEAN(13)
      DIMENSION BLOCK(10).COUNT(10).FROM(10).TO(10)
      DIMENSION DEM(13.25), TDEM(25), WORD(10)
      DIMENSION PERPAS(13.10), PERFAIL(13.10)
      DIMENSION      PTLF(20.5).ETDIST2(13.25)
      DIMENSION  KFD(13).KPD(13)
      INTEGER DEF, PROFILE.PEECFE.AREAS
      INTEGER PROLIM
      FAKE = RNOG(1)
      RUN = 0.0
C     READ PARAMETERS FOR THIS RUN
    3 READ 201.ISTUD.NOET.NO2ET.PROLIM.DEF
      IF(EOF.60) 800.4
C     READ IN NEW VALUES
    4 CONTINUE
      CALL DEFINE(DEF,PROFILE,PEECEE,AREAS)
      RUN = RUN + 1.    $    KP = 0
      STUD =  ISTUD
      READ 209, (WORD(J).J=1.5)
      DO 5 M=1.PROLIM
    5 READ 210,  (     PTLE(M.I).I=1.3)
      READ 203, ((PCINA(I.J).J=1.13).I=1.PROLIM)
      READ 203, (APPF(J).J=1.13)
      READ 203, (APOST(J).J=1.13)
      READ 207, (ETIME(J).J=1.25)
      READ 205, ((ETDIST(I.J).J=1.25).I=1.13)
      READ 205, ((ETDIST2(I.J).J=1.25).I=1.13)
C     HEADING OUTPUT
      KP = KP + 1   $   PRINT 220.KP
      PRINT 302
      PRINT 304, RUN
      PRINT 306, (WORD(J).J=1.5)
      PRINT 308.PROFILE.ISTUD
      PRINT 310.PROFILE.PROLIM
      PRINT 312
      PRINT 230
      PRINT 314, (APRF(J).J=1.13)
      PRINT 315, (APOST(J).J=1.13)
C     CREATE NEW CUMULATIVE E.T. DIST
      DO 17 J=1.13
      DCUM(J.1) = ETDIST(J.1)
      SCUM(J.1)=ETDIST2(J.1)
      DO 15 I=2.NOET
   15 DCUM(J.I) = DCUM(J.I-1) + ETDIST(J.I)
      DO 16 I=2.NO2ET
   16 SCUM(J.I) = SCUM(J.I-1) + ETDIST2(J.I)
C     DO 17 I=1.NOFT
C     SCUM(J.I) = SCUM(J.I) -.001
C  17 DCUM(J.I) =DCUM(J.I)-.001
   17 CONTINUE
C     PRINT 267, ((DCUM(I.J).I=1.13).J=1.NOFT)
C     PRINT 271
      DO 400 IPRO=1.PROLIM
C     CLEAR AREA COUNTERS
      DO 11 I=1.13    $    TSUM(I)=0.
      DO 11 J=1.10    $    PASS(I.J)=0.    $    FAIL(I.J)=0.
   11 CONTINUE
      DO 12 J=1,13
      DO 12 I=1,25   $   TDEM(I) = 0.
   12 DEM(J,I) = 0.
C     LOOP FOR NO. OF STUDENTS
      DO 100 IS=1,ISTUD
C     FOR EACH STUDENT, MARCH THROUGH THE MAZE
C     SUBROUTINE PTEST NOW HANDLES THE AREAS
   25 CALL PTEST(IPRO,KFD,KPD)
C     HAVING COMPLETED PRE/POST, SAMPLE ETDIST
      CALL SAMPLE (KFD,KPD)
      STIME=0.
C     FIND STUDENT TIME AND STORE ON DRUM VIA LU 11
      DO 29 I=1,13  $  CONST=0.  $  DO 27 J=1,NOET
      DEM(I,J) = DEM(I,J) + ETSUM(I,J)
   27 CONST = CONST + ETSUM(I,J) * ETIME(J)
      TSUM(I) = TSUM(I) + CONST
   29 STIME = STIME + CONST
      WRITE(11,251) IS, STIME
      PRINT 253, IS,STIME
C     END STUDENT LOOP
  100 CONTINUE
C     PROFILE STATISTICS SECTION
      REWIND 11
C     SEARCH LIST TWO TIMES FOR HIGH AND LOW
      READ (11,251) IDUM,HIGH
      ALOW = HIGH
      DO 41 I=2,ISTUD
      READ (11,251) IDUM,TEST
      IF(TEST.GT.HIGH) 43,45
   43 HIGH = TEST
      GO TO 41
   45 IF(TEST .LT. ALOW) 47,41
   47 ALOW = TEST
   41 CONTINUE
      REWIND 11
C     SET UP INTERVALS FOR HISTOGRAM
      SUB = 10.
      ISUB = SUB
      TDIF = HIGH - ALOW
      A = TDIF/SUB + .9
      INC = A   $   AINC = INC
      BLOCK(1) = ALOW + AINC - 1.
      DO 61 J=2,ISUB
   61 BLOCK(J) = BLOCK(J-1) + AINC
C     PRINT 301, (BLOCK(I),I=1,10)
      FROM(1) = ALOW
      TO(1) = ALOW + AINC
      DO 65 J=2,ISUB
      FROM(J) = FROM(J-1) + AINC+1.
   65 TO(J) = TO(J-1) + AINC+1.
      DO 48 K=1,10
   48 COUNT(K) = 0.
      T = 0.   $   TSQ = 0.
C     READ IN TIMES FOR HISTOGRAM AND MEAN-SD COMPUTATION
      DO 51 I=1,ISTUD
      READ (11,251) IDUM,TIME
C     COUNT THE TIME IN A HIST INTERVAL
      DO 62 J=1,ISUB
      IF(TIME.LE.BLOCK(J)) 63,62
   62 CONTINUE
   63 COUNT(J) = COUNT(J) + 1.
      T = T + TIME
   51 TSQ = TSQ + TIME**2
      AMEAN = T/STUD
      DENOM = STUD * (STUD-1.)
      VAR = (STUD*TSQ - T**2)/DENOM
      SD = SQRT(VAR)
      DO 53 J=1,13
   53 TMEAN(J)= TSUM(J)/(STUD-PASS(J,1))
      DO 56 J=1,NOET
      DO 56 L=1,13
   56 TDEM(J) = TDEM(J) + DEM(L,J)
C     COMPUTE PASS / FAIL OCCURANCES FOR 1 TO 10 TIMES
      DO 67 I=1,13
      DO 67 J=1,10
      PERPAS(I,J) = PASS(I,J) / STUD
   67 PERFAIL(I,J) = FAIL(I,J) / STUD
      REWIND 11
C     * * * * * * OUTPUT SECTION * *
C     PROFILE PAGES OUTPUT
      KP=KP+1   $   PRINT 220,KP
      PRINT 345, PROFILE,  IPRO
      PRINT 335,  (     PTLE(IPRO,I),I=1,3)
      PRINT 346,PEECEE,PROFILE
      PRINT 230
      PRINT 348,PEECEE,  (PCINA(IPRO,J),J=1,13)
      PRINT 350
      PRINT 230
      DO 71 K=1,NOET
   71 PRINT 352, K, (ETDIST(J,K),J=1,13)
      PRINT 351
      PRINT 230
      DO 70 K=1,NO2ET
   70 PRINT 352, K, (ETDIST2(J,K),J=1,13)
      PRINT 354,  PEECEE
      PRINT 355
      PRINT 230
      DO 72 K=1,10
   72 PRINT 356, K, (PERPAS(J,K),J=1,13)
      PRINT 358, PEECEE
      PRINT 359
      PRINT 230
      DO 73 K=1,10
   73 PRINT 356, K, (PERFAIL(J,K),J=1,13)
      KP=KP+1   $   PRINT 220,KP
      PRINT 375, PROFILE
      PRINT 377, PROFILE,  HIGH
      PRINT 378, PROFILE,  ALOW
      PRINT 380, PROFILE,  AMEAN, SD
C     HISTOGRAM OUTPUT
      PRINT 382, PROFILE, IPRO,INC
      PRINT 384, (FROM(J),J=1,ISUB)
      PRINT 385, (TO(J),J=1,ISUB)
      PRINT 386
      PRINT 388, (COUNT(J),J=1,ISUB)
      PRINT 386
      PRINT 390,  AREAS
      PRINT 230
      PRINT 392, (TMEAN(J),J=1,13)
```

```
382 FORMAT (    47X.*HISTOGRAM FOR*.A5.I5./.   3X.*WITH INCREMENTS OF*.
   1I4.* HOURS*.//)
384 FORMAT (5X.* FROM *.3X.10(F5.0.4X))
385 FORMAT (/.5X.* TO *.4X.10(F5.0.4X))
386 FORMAT (/5X.8H* -- * -.10(9H * - - * ))
388 FORMAT (/.5X.*NUMBER*.5X.10(F3.0.6X))
390 FORMAT (///.37X.*THE AVERAGE TIME IN HOURS SPENT IN EACH *.A5./)
392 FORMAT (5X.*HOURS*.3X.13F8.1)
400 FORMAT (45X.*DEMAND FOR EVENT TYPES-WITH*.I3.* TYPES AVAILABLE*./)
402 FORMAT (5X.*ET*.I2.3X.13F8.0)
404 FORMAT (//.20X.*SUM OF ABOVE DEMAND - BY TYPE*./)
406 FORMAT (25X.10F8.0)
    END
    SUBROUTINE PTEST(IPRO.KFD.KPD)
    COMMON /A/ PCINA(12.13). APRE(13). APOST(13). TPCINA (13).
   1PASS(13.10). FAIL(13.10). DCUM(13.25). ETSUM(13.25). NOET.
   1SCUM(13.25).NO2ET
    DIMENSION KFD(13).KPD(13)
C   SHIFT PCINA TO TPCINA
    DO 13 J=1.13
 13 TPCINA(J) = PCINA(IPRO.J)
    DO 100 IA=1.13
    P = 0.
    K = 0
    NOPC = PCINA(IPRO.IA)
C   PRETEST SECTION
 10 DO 20 I=1.NOPC
    R = RNOG(2)
    IF(APRE(IA)-R) 20.18.18
 18 TPCINA(IA) = TPCINA(IA) - 1.
    K = K + 1
    IF(K.GT.10) K=10
    PASS(IA.K) = PASS(IA.K) + 1.
    P = P + 1.
 20 CONTINUE
    KPD(IA)=TPCINA(IA)
C   POSTTEST SECTION
    F = 0.
    KF = 0
    NOPC = TPCINA(IA)
    DO 30 I=1.NOPC
    R = RNOG(2)
    IF(APOST(IA) - R) 25.30.30
 25 TPCINA(IA) = TPCINA(IA) + 1.
    KF = KF + 1
    IF(KF.GT.10) KF=10
    FAIL(IA.KF) = FAIL(IA.KF) + 1.
    F = F + 1.
 30 CONTINUE
C   PRINT 50. IA. P. F
    KFD(IA)=KF
100 CONTINUE
 50 FORMAT (10X.*FROM SUB TEST  AREA IS *.I5.5X.*PASSING*. F5.0.5X.
   1*FAILURES*.F5.0)
    RETURN   $   END
    SUBROUTINE DEFINE(DEF.PROFILE.PEECLE.AREAS)
    INTEGER DEF. PROFILE.PEECEE.AREAS
    PROFILE=5H
    PFECEF=5H
    AREAS=5H
```

```
      KP=KP+1    1   PRINT 220.KP
      PRINT 400, NOET
      PRINT 230
      DO 74 K=1.NOET
   74 PRINT 402, K, (DEM(J.K).J=1.13)
      PRINT 404
      PRINT 406, (TDEM(J).J=1.NOET)
C     * * * * * END OUTPUT
C     GET A NEW PROFILE
  400 CONTINUE
C     GET A NEW RUN OTHERWISE STOP
      GO TO 3
  800 STOP
C     * * * * * * OUTPUT FORMATS * *
C     * * * * GENERAL FORMATS * *
  201 FORMAT (4I4.2X.A1)
  203 FORMAT (13F5.0)
  205 FORMAT (25F3.0)
  207 FORMAT (5F5.0)
  209 FORMAT (5A8)
  210 FORMAT (2A8.A4)
  231 FORMAT (45X.5A8)
  251 FORMAT (I3.F5.0)
  215 FORMAT (110X.*PAGE NO.*.I3./)
  220 FORMAT (1H1.110X.*PAGE NO.*.I3./)
  230 FORMAT (16X.* NUM.   P. V.    ADD.    SUB.   MULT.    DIV.   C.O.P
     2.  FRACT.  MONEY   TIME    MEAS.    GEO.  SP.TOP.*./)
C     * * * * HEADING PAGE * * * *
  253 FORMAT (2X.I5.3X.F10.4)
  302 FORMAT(58X.*E D S I M  1*.//.36X.*S C H O O L  O F  E D U C A T I
     1O N  S I M U L A T I O N*.6(/))
  304 FORMAT (52X.*RUN NUMBER *.F2.0.///)
  306 FORMAT (30X.*DATA DERIVED FROM *.5A8.//)
  308 FORMAT (20X.*THE NUMBER OF STUDENTS IN EACH *.A5.* IS*.I5./)
  310 FORMAT (20X.*THE NUMBER OF *.A5. *S FOR WHICH RESULTS WILL BE OUTP
     1UTED IS *.I3.//)
  312 FORMAT (42X.*PRE AND POST TEST PROBABILITIES FOR THIS RUN*.//)
  314 FORMAT (7X.*PRE*.5X.13(2X.F5.3.1X))
  315 FORMAT (/.6X.*POST*.5X.13(2X.F5.3.1X))
  335 FORMAT (57X.2A8.A4.//)
C     * * * * PROFILE PAGES * * * * *
  345 FORMAT (50X.*OUTPUT STATISTICS FOR *.A5.I4./)
  346 FORMAT (47X.*NUMBER OF *.A5.*S IN THIS *.A5./)
  348 FORMAT (4X.A5.  3X.13(2X.F5.0.1X))
  350 FORMAT (1H1./.45X.*EVENT TYPE SELECTION PROBABILITIES*./.51X.
     1 *BEFORE POST TEST*./)
  351 FORMAT (1H1./.45X.*EVENT TYPE SELECTION PROBABILITIES*./.51X.
     1*AFTER POST TEST*./)
  352 FORMAT (5X.*ET*.I3.2X.13F8.3)
  354 FORMAT (//.42X.*PERCENTAGES OF STUDENTS PASSING *.A5.* PRETESTS*)
  355 FORMAT (4X.*PERCENT*./.4X.*PASSING*)
  356 FORMAT (2X.*AT LEAST *.I2.1X.13(3X.F5.3))
  358 FORMAT (//.42X.*PERCENTAGES OF STUDENTS FAILING *.A5.* POST TESTS*
     9 )                                                               35d
  359 FORMAT (4X.*PERCENT*./.4X.*FAILING*)
  375 FORMAT (43X.*TIME DIMENSION OF THIS *.A5.//)
  377 FORMAT (20X.*HIGHEST TIME IN THE *.A5. * IS*.F8.1. * HOURS*./)
  378 FORMAT (20X.*LOWEST TIME IN THE *.A5. * IS*.F8.1.* HOURS*./)
  380 FORMAT (20X.*THE AVERAGE *.A5. * TIME IS *.F8.2.* HOURS WITH A STA
     1NDARD DEVIATION OF *.F7.2.* HOURS*.///)
```

```
      IF (DEF.EQ.1HL)  10, 20
   10 PROFILE=5HLEVEL
      PEECEE=5H AREA
      AREAS=5H AREA
      GO TO 30
   20 IF (DEF.EQ.1HA)  25, 30
   25 PROFILE=5H AREA
      PEECEE=5HSKILL
      AREAS=5HSKILL
   30 RETURN
      END
      SUBROUTINE SAMPLE(KFD,KPD)
      COMMON /A/ PCINA(12,13), APRE(13), APOST(13), TPCINA (13),
     1PASS(13,10), FAIL(13,10), DCUM(13,25), ETSUM(13,25), NOET,
     1SCUM(13,25),NO2ET
      DIMENSION KFD(13),KPD(13)
      DO 8 I=1,13  $  DO 8 J=1,25
    8 ETSUM(I,J) = 0.
C     FOR ALL AREAS, WITH NEW TPCINA
      DO 80 J=1,13
      N=TPCINA(J)
C     GENERATE RANDOM NO. AND FIT INTO CUM DIST.
      M=KPD(J)
      DO 50 I=1,M
      R=RNOG(2)
      DO 30  IDIST=1,NOET
       ISAVE=IDIST
       IF(R-DCUM(J,IDIST))25,25,30
   30 CONTINUE
   25 ETSUM(J,ISAVE)=ETSUM(J,ISAVE)+1.
   50 CONTINUE
      MM=KFD(J)
      DO 65 I=1,MM
      R= RNOG(2)
      DO 60 IDIST = 1,NO2ET
       ISAVE=IDIST
       IF (R-SCUM(J,IDIST)) 63,63,60
   60 CONTINUE
   63 ETSUM(J,ISAVE)= ETSUM(J,ISAVE) +1
   65 CONTINUE
   80 CONTINUE
C     PRINT 70, ((ETSUM(I,J),I=1,13),J=1,NOET)
   70 FORMAT (5X,*FROM SUB SAMPLE  ETSUM*,/,(10X,13F5.0))
      RETURN  $  END
      FUNCTION RNOG(ICODE)
C     GENERATES ONE NO. PER PASS OF LENGTH DETERMINDED BY INDEX
      IF (ICODE.EQ.99999999)  GO TO 100
      RNOG=RANF(-1)
  100 CONTINUE
      RETURN  $  END
          SCOPE
*LOAD
```

EDSIM 2

EDSIM 2[1] is the first attempt to create a large scale model. We suggest that the potential user consider EDSIM 4. EDSIM 2 is presented for historic completeness and for the few products not included in later models. Using a class of "created" students, EDSIM 2 the computer simulation "steps through" the same sequence of events as the envisioned METEP design. It then reports a series of statistics on various levels of resource use and rates of progress in the students. The simulation thus provides a means to see the potential impact on the total METEP system of design changes being contemplating. (Figure A)

An initial attempt to create a single computer program to exhibit the essential variety of the METEP system was deemed inappropriate. From a model building standpoint, it became difficult to "step through" the model logically, an essential step if the model is to have any internal validity.

Hence, the simulation model is actually three separate computer programs. Two contain the elements of the METEP, the third is a data translating program. Prior to examination of these programs, a brief description of the data structure in the simulation will be presented.

Much of the simulation model concerns the transmission and updating of data files. Four primary files are needed by the METEP, and consequently by the simulation model. The first file is obviously a file of student status records. This file "tracks" each student through the system. The second file lists the I. A.[1] associated with each P. C.[2] A student who must meet a particular P. C. must be able to find out what I. A. are available to aid him. The first file is derived from the second, it lists the P. C. one is eligible to test given he had engaged in a particular I. A.

---

[1] This model was developed as part of the Model Elementary Teacher Education Project at the University of Massachusetts. Reports of the Project are available through ERIC as follows:

"A Feasibility Study on the METEP" Phase II, Vol. 1. Ed 043-582.

"A Feasibility Study on the METEP" Phase II, Vol. 11. Final Report. Ed 043-583.

"Summary: A Feasibility Study on the METEP" Phase II. Final Report. Ed 043-584.

The importance of this third file may not be apparent. A "rational" student facing a list of P. C. he must pass will not arbitrarily chose I. A. Rather, he will select I. A. in such a manner that a minimum of duplication will be made. A lengthy I. A. may meet 10 P. C. The "rational" student will not take this I. A. to meet just one P. C. and forego the opportunity to test the other nine. It is the third file which allows a complete linking of I. A. to P. C.

Finally, a fourth file emanates from the system administration. It contains the requirements needed to conduct any I. A. The requirements relate to physical space, faculty, and assistants. From such a file, resource requirements for any set of I. A. can be listed.

The simulation model is built around these four files of information. This file structure occupies a dominant position that the organization of the computer programs which comprise the simulation.

Postponing for a moment the first program, which is a data editor, the remaining two programs constitute the actual simulation. They are a student generator, called STUDENT, and an METEP flow program, labeled ANARCHY.[3] There were several practical reasons for "creating" students in a separate program. The most important is the need to have a constant population for all the model runs. If each run is to examine the consequences of a change in the METEP flow, comparison between runs would be very difficult if the student population were altered between runs.

---

[1] An "Instructional Alternative" (IA) is an activity through which a student prepares himself to meet the stated criterion and occasionally several related criteria.

[2] A "Performance Criterion" (PC) is a specification of an instructional goal in terms of behavior to be exhibited or actions to be performed by the student.

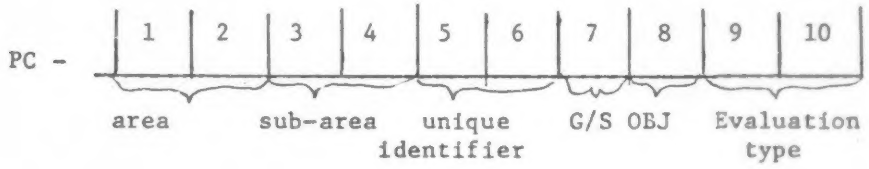
[3] The name was created upon watching the computer's first efforts to digest the program.

Consequently, the program STUDENT is only run to create a student population. This set of students is then used in all subsequent runs of the METEP flow program ANARCHY. An exception occurs when the nature of the P. C. or I. A. are altered. As the STUDENT program selects I. A. for the students, new I. A. necessitates a new set of students with appropriate selections.

Program I - Edit and Translation

This program was written to convert the codes used by the educators into codes suitable for the remaining two programs. In addition to translating the codes, the program also performs a data edit. Input to this program consists of "raw" data on the P. C., I. A., and requirements. The results of the program are an "edited" set of data, a list of errors, and a conversion table. The error list can be used to correct the "raw" data. The connected "raw" data can then be cycled through the program again. This cycling yields the maximum amount of usable "edited" data.

Each P. C., I. A., and requirement must be specified by a code. The educator's codes are given below; where each slot represents one digit.



Explanation=

AREA = which area the P.C. is placed in, i.e., math, human relations, social studies.

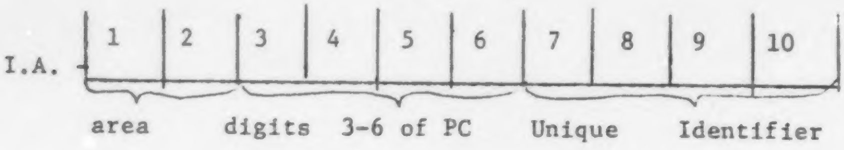
SUB-AREA= a sub-division within the AREA, i.e. addition of fractions in mathematics.

UNIQUE IDENTIFIER: A unique two digit number that separates the P.C. from all others.

G/S = Indicates whether this P.C. must be met by all students or only those specializing in the area.

OBJ TYPE= Code for the objective of the P.C., i.e. to implant a skill, or a behavior pattern.

EVALUATION TYPE = Code for the type of test used to evaluate the student's mastery of the P.C.



Explanation=

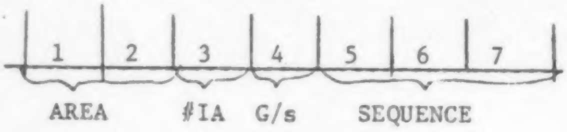
AREA = which area the I.A. belongs in

DIGITS 3-6 OF PC = the digits 3-6 of the PC relating to this I.A. are inserted in the I.A. code.

UNIQUE IDENTIFIER = A unique four digit number which isolates this I.A. form all others.

Program I translates the above codes into the following codes, which are used in the remainder of the simulation.

SIMULATION PC CODE =



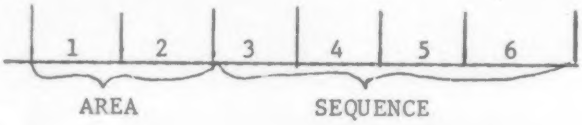
Explanation=

AREA and G/S are as above

#IA = number of I.A. which can be taken in an effort to satisfy the requirements of the P.C.

SEQUENCE = A unique, sequential, number isolating this P.C. from all others. The sequence begins at the number one and works up.

SIMULATION IA CODE =



Explanation =

AREA is as above

SEQUENCE = a unique four digit identifier going in sequence from the number one.

When examining the requirements, the program reads a set of data in each I. A. code. This data consists of ten elements. These elements are shown below.

IA REQUIREMENT ELEMENTS

ELEMENT 1 = the 10 digit I. A. code to which the following elements will relate.

ELEMENT 2 = the percent per day this I. A. will consume. The figure is based on a 10 hour day. Thus 10% would be 1 hour.

ELEMENT 3 = duration of the I. A. in days.

ELEMENT 4 = minimum number of students necessary to initiate this I. A.

ELEMENT 5 = maximum number of students that can participate in this I. A.

ELEMENT 6 = the type of faculty needed, if any.

ELEMENT 7 = the number of hours this faculty member must commit per week to this I. A.

ELEMENT 8 = a second faculty type needed, if any.

ELEMENT 9 = the hours per week this second member must commit.

ELEMENT 10 = the type of physical resource required, such as a classroom or a micro-teaching lab.

Program I alters only three of the above elements. Altered elements are:

ELEMENT 1 = now contains the translated 6 digit I. A. code.

ELEMENT 2 = is translated from per cent of a day into hours per week.

ELEMENT 3 = is translated from number of days duration into number of weeks.

Program I consumes the raw data in the following form. The P. C.-I. A. file is created by the educator using his codes. This file contains the P. C. code and all the I. A. codes that link to this P. C. This file is placed on data cards, one P. C.-I. A. set per card. A limit of four I. A. linking to one P. C. is imposed. Program I creates from the P. C.-I. A. file the I. A.-P. C. file containing the P. C. associated with each I. A. In this file, only 6 P. C. are allowed to link to one I. A. Thus the educator need not concern himself with creating the I. A.-P. C. file.

The requirements enter Program I as one set of ten elements per data card. Hence one data card is needed in each I. A. Program I converts these two files of data cards to three files of data cards with "edited" codes. The process is represented in Figure B.

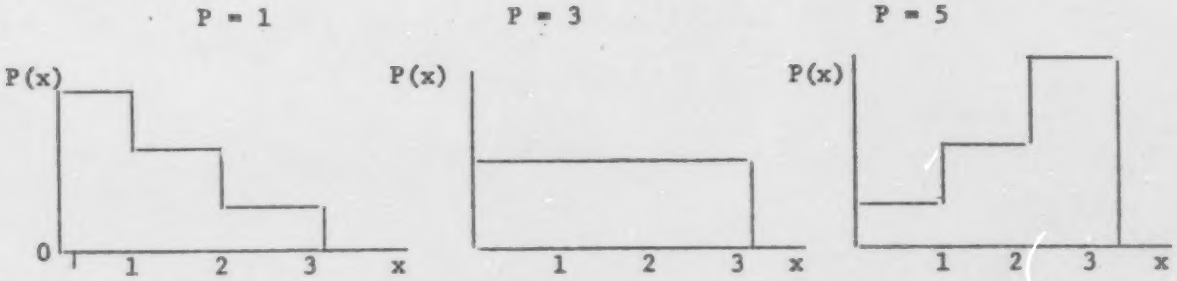
Program II - Student Creation

Utilizing the data translated by Program I, Program II "created" a set of students suitable for the METEP system. Each suitable student has a record which contains the complete list of P. C. he will have to meet to graduate from the METEP system. The students and their records are generated by the following scheme.

At the time, the METEP system contained about 580 P. C., the majority of which must be met by any student in the system. The set of P.C. which will be tested by every student are known as generalist P. C. Each area has at least one generalist P. C. Because the specialist portion of the METEP is not fully developed, only generalist P. C. were included in the data in the simulation model.

Hence, each student upon entering the METEP system faces a given set of generalist P. C. He may elect to pretest the P. C., as one of the METEP tenets is that a student should not have to labor through an I. A. if he can demonstrate the proficiency required by the related P. C. Program II assumes that a student will elect to attempt to pretest each P. C. Those P. C. successfully pretested do not require further action.

For those P. C. not successfully pretested, an I. A. must be selected with the aid of the P. C.-I. A. file previously described. The selection process is set up so that any one of a set of I. A. linking to this P. C. is equally likely to be chosen. Thus if a student can select from three I. A., the selection process assigns a probability of 1/3 to each I. A. in the set. Program II has the ability to set non-equally likely probabilities for I. A. selection. This is done using the input variable PRAMOST (see run details), and an element of the student record called PARAMETER. The PARAMETER shifts the probability distribution from equally likely to greater likelihood of selecting the first or last I. A.'s of a set. For example, if three I. A.'s are in a set, the PARAMETER values 1, 3, 5 would yield probability distributions:



For all the current work, the parameter value is set at 3, giving the equally likely distribution.

Having selected an I. A., the file I. A.-P. C. is utilized to check off any other P. C. which can be met using this I. A. This avoids over-selecting I. A. Such a situation would not be congruent with the METEP system.

The above routine is repeated until each P. C. in the generalist category has been dealt with. The resulting list of I. A. comprises those I. A. this student must complete during his stay in the METEP system. The I. A. are recorded using the six digit code.

Prior to completing action on this student, Program II completes two other operations. First, each six digit I. A. codes is expanded to a 14 digit INTERNAL I. A. CODE, shown below:

1	2	3	4	5	6	7	8	9	10	11	12	13	14
6-DIGIT CODE						STATUS	Hrs/wk		NO. WEEKS			COUNTER	

Explanation=

STATUS : used by Program III, explanation is defined until then.

HRS/WK: the hours per week consumed by the IA.

NO. WEEKS: number of weeks this IA takes for completion

COUNTER: used by Program III, explanation is defined until then.

After expansion of the code, Program II inserts the hours per week and the number of weeks associated with this IA. The data for this comes from the IA requirements file mentioned earlier. In the second operation, all the IA in the student's list are sorted into areas, the areas running from one upward.

A completed student appears as a list of 14 digit internal IA codes plus some identifiers. This record, called STREC, looks like:

STREC ELEMENTS

ELEMENT (1) : Student I.D.
A numeric code identifying each student

ELEMENT (2) : ENTRY POINT
Used in Program III, explained later

ELEMENT (3) : PROFILE
Currently only a generalist profile is used

ELEMENT (4) : PARAMETER
Identifies the probability distribution used in selecting the IA along with PRAMOST

ELEMENT (5) : STUDENT HOURS PER WEEK
used in Program III, explained later

ELEMENT (6) to ELEMENT (10)
Used in Program III, explained later

ELEMENT (11) : FIRST 14 DIGIT IA

ELEMENT (N) : LAST 14 DIGIT IA

One set of STREC elements is generated for each student. The STRECs are stored by Program II on magnetic tape for use by Program III. The data configuration for Program II is shown in figure C. The input to Program II shown in figure C is reproduced in detail in figure D with an explanation given below. Program II structure is graphically sketched in figure E.

element 1 - NUMPRO
number of profiles, currently limited to one, the generalist (14)

element 2 - NUMAREA
number of areas in the system (for METEP this is 11) (14)

element 3 - NUMSTUD
number of students to be created (14)

element 4 - NRPC
number of generalist PC in the PC-IA file (14)

element 5 - P CIAR ARRAY
the PC-IA file in coded form (I7, 3X, 4[I6,4X])

element 6 - PRETEST ARRAY
the pretest probability for each of the areas, used to set the pretest level for the student

element 7 - PRAMOST ARRAY
The array of probability distributions used in IA selection (F4.0)

element 8 - NIAX
total IA in the IA-PC and requirements file (14)

element 9 - IAREQ ARRAY
the IA requirements file (I6, 8X, 2I2, 6X, I2, I3)

element 10- IAPCR ARRAY
the IA-PC linking file (I6,4X, 8 [I7,1X])

The notation in parentheses refer to the FORTRAN language imput formats associated with each data element.

ANARCHY

Program III, labeled ANARCHY, is the most complex of the three programs. Programs I and II are, in effect, preliminary steps to running Program III. It is in Program III that the essential operating characteristics of the METEP are simulated.

Program III consists of a main program and several sub-programs referenced by the main program. Figure F is a table of the program names and a cursory description of their activity. Program III is cyclic in construction, so that the sub-programs are called many times under varying conditions. Data is transfered between sub-programs via common statements and movement of data between sub-programs is taken to be automatic in all that follows.

With the aid of Figure G, which is a flowchart of the basic program mechanics, the actions of the program can be sketched.

The first step is to input all the data necessary for the run. Primarily, this consists of parameter settings, requirements, data and resource data. All data input except student records is handled by the DATAREAD subroutine. A complete list of input cards is found in Figure H, which also contains a brief description of each data element. The descriptions will be elaborated upon where necessary.

ANARCHY is a cyclic program with a major cycle for each semester to be simulated. Each semester a new group of students can enter the sytem. The number of students to enter in any given semester is found from the NUMSTUD array. For each entering student record, ten pieces of information are added to the record. In element 2 goes the current semester number, indicating the entering semester. In element 5 is put the hours per week the student will be willing to commit. This number is obtained from the subprogram HOURS, utilizing a probability distribution found in the input array STIMDST. Upon gaining these two elements, the student record is opened to the file of student records currently in the system.

ANARCHY is now ready to step through a semester. Two phases are employed. In the first, each student record is allocated 20 slots of an array holding the student's active elements. In the first ten slots of the array, called CURRNT, as the students' ID and his hours per week. Next each record is searched for any unscheduled semester length IA. Any found are entered into the first positions of the students' CURRNT.

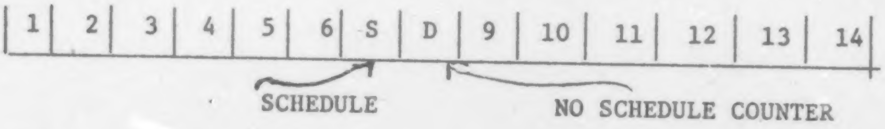
The IA in the CURRNT array are then summed into a demand list. The demand list, IADEM, holds the IA demanded and the number of requests for a given IA. At this point IADEM contains only semester length requests.

Processing the requests involves three sub-programs, MATCH 1, MATCH 2, and CURPACK. MATCH 1 matches the requests found in IADEM against the available resources. For semester length IA, for steps are involved. Recall that each IA lies in an area. Only a certain percentage of an area's resource can be committed to semester length IA. The percentage is an input element, TMAX. So each IA in the demand list must be tested for exceeding allowable resources.

Assuming an IA does not exceed the limit, it in effect is "scheduled", resources are committed to the IA, and a code indicating "OK" is placed next to the IA in the demand list. If the allowable value is exceeded the IA is not "scheduled", no resources are committed, and a code indicating "not scheduled, exceeds allowable range" is placed in the demand list. Each IA is processed in turn, so that prior commitments of resources do effect the ability to "schedule" IA further down the list.

The demand array IADEM now goes to MATCH 2. This sub-program uses the schedule codes to update the IA in the CURRNT array. To each IA in CURRNT is added an indicator in "schedule" or "no scheduled". If the IA has not been scheduled, a no-schedule-counter in the IA is increased by one. The additions are made to the 14 digit IA code previously described.

14 DIGIT IA CODE - ELEMENTS USED IN MATCH-2



The seventh position of the 14 digit code holds the schedule indicator. If the IA was not scheduled, the sum in the eighth position is increased by one, up to a maximum of eight.

CURPACK removes from the CURRNT any unscheduled semester length IA and "packs down" each students' 20 elements so that there are no gaps between IA in the set.

At this point the CURRNT vector holds those semester length IA which have been scheduled. Phase two, the weekly cycle is now ready to commence.

The weekly cycle is performed once for each student. His complete record is made available. The IA in his CURRNT now are summed by time needed per week. The resulting sum is subtracted from the students hours per week, found in element 2 of the CURRNT row. If the difference is positive, there is room to demand additional IA. A test is made on the size of this positive figure. It must be greater than the input element SCHLOW if any further demands are to be made this week.

With room to demand IA, a search of the complete student record is made to pick up additional IA. Two search procedures are used. If the student has completed less than 60% of his IA, then an area is selected at random and IA are pulled from this area until the area is exhausted or until no further IA are needed. If the area is exhausted, another is picked. If greater than 60% of the IA are complete, the selection goes in sequence from the first available IA in the record. This prevents an undo thrashing through areas as the completion nears 100%.

For each IA made available, it is first determined if there are enough weeks to offer the IA, as it is not desirable to demand an eight week IA with only 7 weeks remaining in the semester. If there are enough weeks, the IA will be added to the students CURRNT row if the hours per week required by the IA does not push the student over his own hours per week limit plus a fudge factor. The fudge factor which keeps this scheme flexible, is the input element SLOP.

With an additional IA in CURRNT and hence in the demand list, the new time difference is examined. If the new time is within plus or minus SLOP of the students' hours per week, the demand process is stopped for this student. IF not another IA is pulled from the student record.

The process is repeated for each student. Additional IA are placed in CURRNT and the demand list. Recalling that the CURRNT row holds 20 elements per student, only 17 are available in IA storage. Thus only 17 IA can be in progress or demanded in any one cycle.

The demands for this weekly cycle are placed in IADEM. The program then transfers to MATCH 1. Like the semester length IA, MATCH 1 attempts to match the demands in IADEM with available resources. Three checks are made in each IA, enough students, sufficient faculty time, and sufficient space. A negative result from any check makes the IA a "no schedule" and the proper code is placed in IADEM. A fourth check involves excess demand, that is, more students requesting the IA than the IA can handle. In this case the IA is scheduled and the number of requests is replaced by the maximum number.

For each IA "scheduled" MATCH 1 performs the necessary bookkeeping on available resources. With long demand lists and finite resources, many IA are not scheduled due to lack of resources.

MATCH 2 is again used to transfer results from IADEM to the CURRNT array. CURRNT hold the results of this weeks scheduling efforts.

The main program, ANARCHY, now increases the weekly counter and checks the CURRNT for any completions. If an IA is completed, it is assumed that the student will successfully pass any posttest associated with the IA. Thus, an IA which is completed in the time dimension is assumed completed in all aspects. A code denoting completion of an IA is placed in the seventh position of the 14 digit IA code. During this completion check, a sum is kept of completions and unscheduled IA.

The complete student records are now updated with the information contained in the CURRNT array. In order not to destroy previous information, a two master system is employed. "Last weeks" complete student records are contained in the "old" master file. A record is taken from the file updated with the new information, and then placed in a "new" master file. At the end of the process, the "new" master file contains the records current to this week. Next week, the "new" master becomes the "old" master, and the cycle repeats itself.

During the updating process a check in graduation is made. Element 6 of the student record holds the total IA in the record. Each week the total completions are compared with a percentage of the total IA.

For most runs, this percentage was 95%. A student is graduated when his completions exceed 95% of his total IA. Graduation involves placing relevent portions of the student record in a separate file for display at the end of the semester. In addition, the record is flagged to be removed from the CURRNT array and the "new" master.

The CURPACK sub-program is called to pack the CURRNT array and to remove any graduated students.

Prior to commencing the next weekly cycle, statistics in the week are computed and printed. The program then returns to the beginning of the weekly cycle. If the week is the last in the current semester, additional statistics are printed from MATCH1 and the semesters graduates are listed along with a status report of all others in the system. The program will return to the beginning of the semester cycle and prepare for a new entering class.

In addition to the input shown in Figure H. one additional card is used. This card governs the restart procedure. Because run times are long, a means of restarting the program was deemed necessary. At the end of each semester the "new" master is copied onto a magnetic tape. To restart the system this tape is feed into the program prior to entering the new semester routine. At most part of one semester is lost from an aborted run. The restart card is the first card of the input deck. The complete input deck is shown in Figure I.

The above discussion should give a feeling for the program's structure. For further study, a detailed set of flow charts is displayed in Figure J. The next section will investigate a sample run and show what types of output are available.

Example,

Perhaps the best way to understand the simulation model is to examine its output. The example is from one of the later runs. The input in the run is given in Figure H , by way of explanation the following is given.

The restart procedure is set in a new run. If this run was a restart run, the second figure would be a 1 and the first would be one less than the starting semester.

Three semesters are to be simulated, with 400 students entering the first semester, none the second and third semesters.

The over-demand variable SCHST has a value 1, but this is just a place holder as this variable is not currently used.

A minimum of 2 free hours is needed if additional IA are to be demanded and ± 5 hours from the students' hours-per-week is considered acceptable limits.

The student hours-per-week distribution governs selection of 20, 25, 30, or 35 hours a week as the student time. Note the distribution is CUMULATIVE. If looked at non-cumulative, it assigns a probability .15 for selecting 20 or 35, and a probability .35 for selecting either 25 or 30.

TMAX value at .5 allows only 50% of the area's resources to be committed to semester length IA.

There are 948 IA in the requirements file and 15 types of physical resources.

Next is listed the capacity of each type of physical resource.

The very lengthy requirements file follows, one card in each IA (only the first and last cards of the file are shown).

14 weeks constitute a semester in the system.

The available physical resources in each week in each type are given in the ICRES array. One card per resource type, 14 values per card.

There are 3 types of faculty and 11 areas. NTEACH holds the faculty hours per week in each type by area. Thus the first three cards give the hours in area 1 in the three faculty types.

The remaining arrays hold alphanumeric titles to give some meaning to the data returned by the program. IWDRES holds labels for the physical resources, EWDAREA identifies the areas, and EWDINS lists the faculty types.

The last array, IWDIS, holds the reasons for not being able to schedule an IA. The first, TMAX, notes that an IA of semester length used greater than 50% of the available resources. The last, EXCESS DEMAND, labels rejections due to too many requests for an IA to handle. In many cases the number of students requesting an IA exceeds the maximum number of students the IA can accomodate.

With input now complete, attention can now be turned to the program's output.

Figures (1) and (2) are partial samples of the initial resource conditions. The available physical resources are drawn from the input arrays ICRES and NTEACH. Any IA scheduled will draw down these resources. In each row there are 14 figures, one in each week of the semester. Figure (2) has only one faculty type, the remaining two types are not shown, but the output is similar.

Figure (3) gives a typical weekly output. The first figure in the weekly output should be self explanatory. Length of demand array shows that 250 distinct IA were demanded by the 400 students in the system. The third figure will be meaningful if the nature of the CURRNT array is recalled. Each row of the CURRNT has 17 positions in holding IA. If a student fills all 17 positions and his time still allows more IA, the next selection will overflow his CURRNT row. The number of students who are stopped this way is the number of CURRNT overflows.

Number of excessive searches checks on the area selection procedure cycling more than 10 times. The two part selection routine explained earlier keeps this figure at 0 in most cases.

The five averages are taken over all students in the system this week. Free-time hours are hours students wished to commit but could not due to unscheduled IA. The number can go negative due to the SLOP factor used in computing student time available.

Disappointments refer to IA demanded but not scheduled. In this case the average student had 1.7 IA demanded but not scheduled. The average hours committed is 27.4 and is very close to the expected value, 27.5 hours, derived from the hours-per-week distribution.

The list of "scheduling hand-ups" gives the number of IA unscheduled due to the reason listed. Inadequate demand occurs if less than the minimum number of students requests an IA. That is, a large lecture will not be held for one or two students.

The last figure is not related to the simulation output. It is the computer time actually used to simulate the week. This figure was added when excessive times were noticed in the later stages of the runs. The real time value is a function PRIMARILY of the number of students in the system, here 400, a high figure.

A set of statistics similar to Figure (3) is produced for each simulated week. In addition, all but the real time figures are punched onto cards for use in separate analysis routines.

The end of the semester brings forth a set of summary statistics. Figure (4) shows the first set, the physical resources used each week. As can be seen, there are very few cases when the resources were exhausted in a given week. This is reflected in Figure (5), a summary report showing mean use and its standard deviation. The figures under AVERAGE give the average per cent utilization of the resource over this semester. Figures (6) and (7) give a similar picture for the first faculty type (professional). From such a picture it can be seen that latter weeks tend to draw more heavily upon faculty in some areas, a trend that was marked in several runs.

The semester's accumulation of "unscheduled IA" reasons is presented next in Figure (8). Unlike the tabulation in the weekly output, this table displays the number of cases in each category. Thus in the first week 181 cases of excess demand were observed. Such a figure might indicate too low a maximum level in the IA in the area.

Figure (9) displays one form of the semester status list, the final table of the semester output. When there are no graduates, only a list of key elements is given. The list is more useful for debugging the system than in stating detailed reports on each student's progress. During the debugging phase, the middle three elements were used to track net progress and the buildup of total disappointments. Such tracking aided in part in validating the program's routines.

When there are graduates, an additional status list appears, of which Figure (10) is an example. This list, which is taken from a different run, lists key elements of the students' record at graduation. Going across the list, the elements are:

LIST NUMBER
STUDENT I.D. NUMBER
ENTERING SEMESTER
EXITING SEMESTER
NO. WEEKS IN THE SYSTEM
PROFILE NUMBER (always 1)
STUDENT'S HOURS/WEEK COMMITMENT
TOTAL IA IN RECORD
TOTAL DISAPPOINTMENTS INCURRED

In this particular case it is interesting to note that the graduating students collected almost as many disappointments as they had IA to meet. Such a condition has major implications for student acceptance of the METEP.

A set of statistics like those given in Figures (1) to (10) appears in each simulated semester. The figures are outputed at the end of each simulated semester, they are not held internally by the computer until the end of the run. This procedure prevents an abnormal termination of the run from destroying all the information accumulated up to that point.

Figure 1

```
                 RESOURCE LOADINGS FOR RUN

                   AVAILABLE CLASSROOMS

                       TYPE X WEEKS

          CAPACITY X HOURS AVAILABLE PER WEEK


                        AUDITORIUM

1800  1800  1800  1800  1800  1800  1800  1800  1800  1800  1800

                      LARGE CLASSROOM

2400  2400  2400  2400  2400  2400  2400  2400  2400  2400  2400

                      MEDIUM CLASSROOM

2400  2400  2400  2400  2400  2400  2400  2400  2400  2400  2400

                        SEMINAR ROOM

2400  2400  2400  2400  2400  2400  2400  2400  2400  2400  2400

                          LIBRARY

4800  4800  4800  4800  4800  4800  4800  4800  4800  4800  4800

                           OFFICE

 360   360   360   360   360   360   360   360   360   360   360

                      MATHEMATICS LAB

1800  1800  1800  1800  1800  1800  1800  1800  1800  1800  1800

                        SCIENCE LAB

1800  1800  1800  1800  1800  1800  1800  1800  1800  1800  1800

                      SOC STUDIES LAB

1800  1800  1800  1800  1800  1800  1800  1800  1800  1800  1800

                      SKILLS DEVE LAB

1800  1800  1800  1800  1800  1800  1800  1800  1800  1800  1800

                       HUMAN REL LAB

2400  2400  2400  2400  2400  2400  2400  2400  2400  2400  2400
```

Figure 2

AVAILABLE INSTRUCTION

AREA X WEEKS X TYPE

NO. FACULTY MEMBERS X TEACHING LOAD

PROFESSIONAL

HUMAN RELATIONS

45 45 45 45 45 45 45 45 45 45 45

MATHEMATICS

45 45 45 45 45 45 45 45 45 45 45

LANGUAGE ARTS 1

30 30 30 30 30 30 30 30 30 30 30

LANGUAGE ARTS 2

15 15 15 15 15 15 15 15 15 15 15

LANGUAGE ARTS 3

15 15 15 15 15 15 15 15 15 15 15

LANGUAGE ARTS 4

15 15 15 15 15 15 15 15 15 15 15

LANGUAGE ARTS 5

15 15 15 15 15 15 15 15 15 15 15

SOCIAL STUDIES

60 60 60 60 60 60 60 60 60 60 60

EARLY CHILDHOOD

45 45 45 45 45 45 45 45 45 45 45

SCIENCE

60 60 60 60 60 60 60 60 60 60 60

Figure 3

```
              OUTPUT FOR WEEK  1 OF SEMESTER NUMBER  1


NUMBER OF GRADUATES   0

LENGTH OF STUDENT DEMAND ARRAY   250

NUMBER OF CURRENT OVERFLOWS   51

NUMBER OF EXCESSIVE SEARCHES   0

AVERAGE NUMBER OF IA UNDERTAKEN  10.20

AVERAGE NUMBER OF FINISHES   8.256

AVERAGE FREE TIME-HOURS   0.585

AVERAGE HOURS COMMITTED   27.379

AVERAGE NUMBER OF DISAPPOINTMENTS   1.709

THIS WEEKS SCHEDULING HANG-UPS

     SEM-LENGTH EXCESS              0

     INADEQUATE DEMAND             29

     INADEQUATE TEACHERS            3

     NO CLASSROOM SPACE             0


WEEKS CYCLE TIME IN MILLISEC. =     116020
```

Figure 4

RESOURCE MATRIX

RESOURCES SCHEDULED DURING SEMESTER-FIRST LINE

RESOURCES NOT SCHEDULED DURING DEMESTER-SECOND LINE

AUDITORIUM

0	300	300	0	300	600	0	0	600	1800	1800	1800	1800
1800	1500	1500	1800	1500	1200	1800	1800	1200	0	0	0	0

LARGE CLASSROOM

1620	1620	962	1143	728	1383	1210	1025	1225	489	1024	856	739
780	780	1438	1257	1672	1017	1190	1375	1175	1911	1376	1544	1661

MEDIUM CLASSROOM

1834	1522	1720	2286	2400	1938	2269	1661	1682	1114	1743	1549	1316
566	878	680	114	0	462	131	739	718	1286	657	851	1084

SEMINAR ROOM

813	1029	1145	1189	1267	1352	1508	1773	1438	1487	1526	1507	2381
1587	1371	1255	1211	1133	1048	892	627	962	913	874	893	19

LIBRARY

1226	1196	1287	1288	1353	1390	1439	1451	1413	1366	2251	2274	2455
3574	3604	3513	3512	3447	3410	3361	3349	3387	3434	2549	2526	2345

OFFICE

152	126	137	109	148	106	101	125	113	100	108	84	94
208	234	223	251	212	254	259	235	247	260	252	276	266

MATHEMATICS LAB

672	532	605	610	622	666	852	538	600	623	827	577	629
1128	1268	1195	1190	1178	1134	948	1262	1200	1177	973	1223	1171

SCIENCE LAB

216	180	227	209	321	319	257	346	151	165	137	160	185
1584	1620	1573	1591	1479	1481	1543	1454	1649	1635	1663	1640	1615

Figure 5

```
RESOURCE STATISTICS FOR SEMESTER NUMBER  1

    RESOURCE           CAPACITY    HIGH     LOW     MEAN      STD DEV      AVERAGE

   AUDITORIUM              300     1800       0      750      732.66        0.42
  LARGE CLASSROOM           60     1620     489     1076      313.73        0.45
 MEDIUM CLASSROOM           30     2400    1114     1735      373.64        0.72
   SEMINAR ROOM             15     2381     813     1409      355.24        0.59
     LIBRARY               200     2455    1196     1567      408.61        0.33
      OFFICE                 3      152      84      116       19.29        0.32
 MATHEMATICS LAB            30      852     532      641       89.59        0.36
   SCIENCE LAB              30      346     105      212       71.22        0.12
 SOC STUDIES LAB            30      857     332      566      134.47        0.31
 SKILLS DEVE LAB            30      996     398      662      180.07        0.37
 HUMAN REL LAB              15     1216     404      874      258.58        0.36
 EARLY CHILD LAB            30     1777     646     1394      337.32        0.77
   LANG ARTS LAB            30     1902     877     1316      337.09        0.37
 MICROTEACH LAB             15      165     135      139        7.55        0.12
  FIELD TEACHING           800     4186    1696     3925      627.81        0.41
```

Figure 6

TEACHER MATRIX

```
           RESOURCES SCHEDULED DURING SEMESTER-FIRST LINE

         RESOURCES NOT SCHEDULED DURING SEMESTER-SECOND LINE

                           PROFESSIONAL

                          HUMAN RELATIONS

 13     31     30     33     23     21     30     41     40     31     44
 32     14     15     12     22     24     15      4      5     14      1
                           MATHEMATICS

 30     25     15     20     38     23     38      3     13     13     44
 15     20     30     25      7     22      7     42     32     32      1
                          LANGUAGE ARTS 1

  4      9     15     10     15     20     10     15     29     20     30
 26     21     15     20     15     10     20     15      1     10      0
                          LANGUAGE ARTS 2

 10     10      5     10     10     10      0      5      0      5      0
  5      5     10      5      5      5     15     10     15     10     15
                          LANGUAGE ARTS 3

 10     10     10     10     10     10     10     10     10     10     10
  5      5      5      5      5      5      5      5      5      5      5
                          LANGUAGE ARTS 4

  5      5      5      5      5      5      5      5      0      0      5
 10     10     10     10     10     10     10     10     15     15     10
                          LANGUAGE ARTS 5

  8      8      0     15     15      9      9      1      0      0     15
  7      7     15      0      0      6      6     14     15     15      0
                           SOCIAL STUDIES

 14     10     14     13     10     13     16     16     10     20     14
 46     50     46     47     50     47     44     44     50     40     46
                          EARLY CHILDHOOD

 15     32     40     39     42     43     43     45     45     43     43
 30     13      5      6      3      2      2      0      0      2      2
                             SCIENCE
```

Figure 7

INSTRUCTION STATISTICS FOR SEMESTER NUMBER 1

RESOURCE	HIGH	LOW	MEAN	STD DEV	AVERAGE
HUMAN RELATIONS	45	13	32	9.18	0.73
MATHEMATICS	44	3	25	11.38	0.56
LANGUAGE ARTS 1	30	4	18	8.43	0.62
LANGUAGE ARTS 2	10	0	5	4.23	0.33
LANGUAGE ARTS 3	10	10	10	0.00	0.67
LANGUAGE ARTS 4	5	0	3	2.25	0.26
LANGUAGE ARTS 5	15	0	5	6.07	0.38
SOCIAL STUDIES	20	8	13	2.98	0.22
EARLY CHILDHOOD	45	15	40	7.73	0.89
SCIENCE	30	8	22	8.46	0.37
BEHAVIOR	11	8	9	1.10	0.63

Figure 8

DISAPPOINTMENT FILE

HUMAN RELATIONS										
TMAX	0	0	0	0	0	0	0	0	0	0
INS DEMAND	0	17	14	29	21	42	40	23	43	62
LACK TEACHER	0	0	0	0	0	0	0	0	0	0
LACK RESOURCE	0	0	0	0	0	0	0	0	0	0
EXCESS DEMAND	181	256	186	218	239	226	208	227	267	225
MATHEMATICS										
TMAX	0	0	0	0	0	0	0	0	0	0
INS DEMAND	28	52	127	121	75	152	134	234	200	236
LACK TEACHER	0	0	0	0	0	0	0	0	0	0
LACK RESOURCE	0	0	0	0	0	0	0	0	0	0
EXCESS DEMAND	0	0	0	0	0	0	0	0	0	0
LANGUAGE ARTS 1										
TMAX	0	0	0	0	0	0	0	0	0	0
INS DEMAND	24	27	56	50	52	69	63	83	58	90
LACK TEACHER	0	0	0	0	0	0	0	0	1	0
LACK RESOURCE	0	0	0	0	0	0	0	0	0	0

Figure 9

```
                  END OF SEMESTER      1

               STUDENT STATUS DUMP

STUDENT    COMPLETIONS    TOTAL   DISAPPOINTMENTS   HOUR AVAIL.
  I.D.       TO DATE                  TO DATE

    1           96         459          25              25
    2          116         453          16              25
    3           81         459          13              20
    4          135         455          25              25
    5          110         464          27              25
    6          148         460          22              30
    7          159         482          22              30
    8          144         458          37              35
    9          114         462          15              25
   10          128         462          13              25
   11          117         462          16              25
   12          118         460          16              25
   13          109         467          34              30
   14           90         461          27              25
   15          115         466          32              30
   16          146         447          39              35
   17          102         462          25              25
   18           60         471          22              20
   19           97         457          14              25
   20          112         451          16              25
   21          133         471          18              35
   22           87         466          14              20
   23          139         456          28              30
   24          139         460          35              35
   25          111         457          33              30
   26          106         460          41              30
   27          148         452          39              35
   28          132         474          20              35
   29          131         449          30              30
   30          148         459          20              35
   31          124         469          33              30
   32          133         457          23              25
   33           95         472          18              20
   34           77         462          19              25
   35           96         468          25              25
   36          103         467          18              30
   37          114         466          23              30
   38          101         467          17              30
   39          149         458          19              35
   40          134         454          33              35
   41          107         469          39              30
   42          123         448          23              30
   43          128         460          50              30
```

Figure 10

```
                    E N D   O F   S E M E S T E R     6

THIS SEMESTERS GRADUATES

  1      30      1      6      80      1      35      459      495
  2      33      1      6      80      1      30      472      457
  3       1      1      6      81      1      30      459      460
  4       4      1      6      81      1      30      455      509
  5       7      1      6      81      1      35      482      579
  6      15      1      6      81      1      35      466      512
  7      38      1      6      81      1      35      467      555
  8      39      1      6     .81      1      35      458      537
  9      72      1      6      81      1      35      451      457
 10      77      1      6      81      1      35      459      539
 11      78      1      6      81      1      35      458      448
 12      85      1      6      81      1      35      467      493
 13      87      1      6      81      1      30      467      477
 14      91      1      6      81      1      35      448      555
 15     113      1      6      81      1      30      465      493
 16     131      1      6      81      1      30      461      408
 17     140      1      6      81      1      35      448      544
 18      12      1      6      82      1      30      460      402
 19      18      1      6      82      1      30      471      428
 20      43      1      6      82      1      35      460      521
 21      49      1      6      82      1      35      478      485
 22      51      1      6      82      1      30      468      429
 23      64      1      6      82      1      35      465      616
 24      79      1      6      82      1      35      469      471
 25      82      1      6      82      1      35      461      422
 26      84      1      6      82      1      30      466      435
 IM     189      1      6      82      1      35      448      572
 28      33      1      6      82      1      35      472      592
 29       8      1      6      83      1      35      458      593
 30      26      1      6      83      1      35      460      595
 31      29      1      6      83      1      35      449      535
 32      42      1      6      83      1      35      448      588
 33      46      1      6      83      1      30      458      466
 34      70      1      6      83      1      30      479      458
 35      73      1      6      83      1      30      463      438
 36      88      1      6      83      1      30      458      478
 37     122      1      6      83      1      30      463      490
 38     134      1      6      83      1      35      467      543
 39     137      1      6      83      1      35      458      573
 40       4      1      6      83      1      35      455      594
 41      19      1      6      83      1      35      457      630
 42      30      1      6      83      1      35      459      534
 43       3      1      6      84      1      30      459      454
 44       6      1      6      84      1      35      460      459
 45      10      1      6      84      1      30      462      333
 46      11      1      6      84      1      30      462      459
 47      22      1      6      84      1      30      466      491
 48      35      1      6      84      1      30      468      484
```

Figure A

SIMULATION DATA FLOW

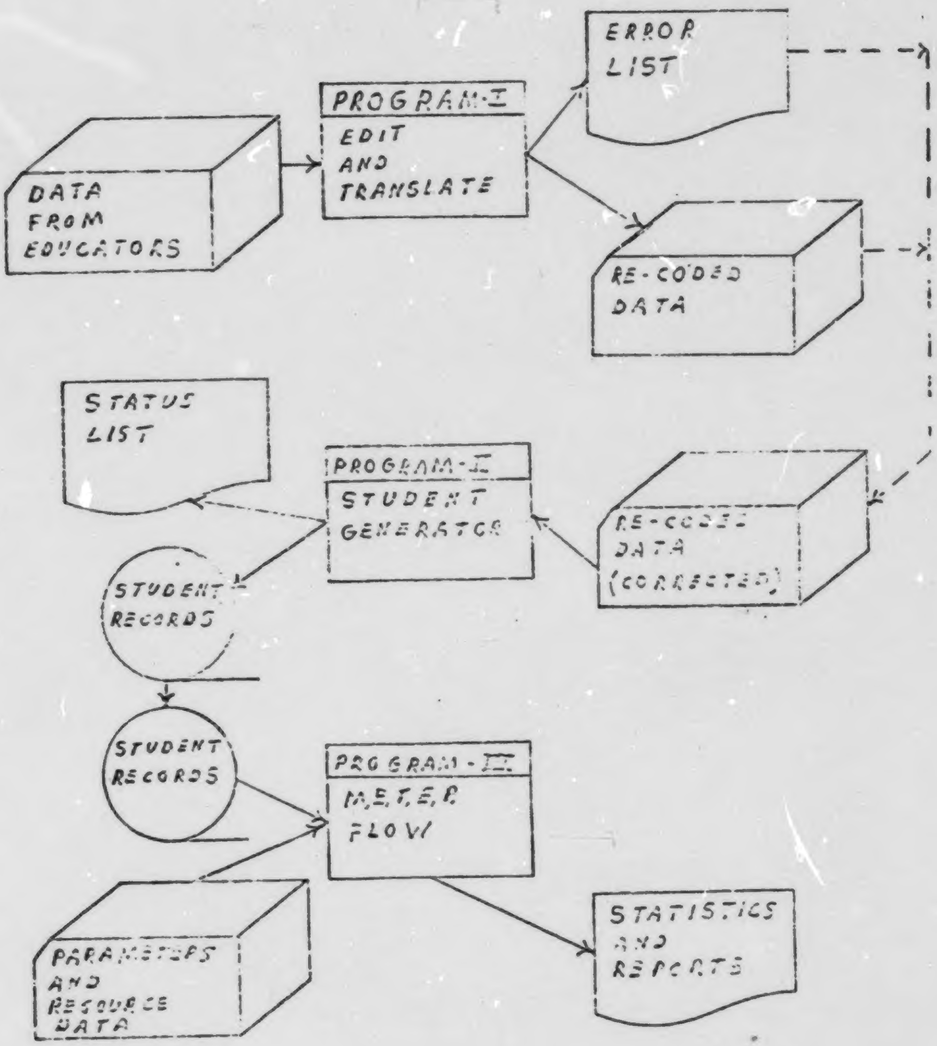

PROGRAM-I
EDIT AND TRANSLATE
ERROR LIST
DATA FROM EDUCATORS
RE-CODED DATA
STATUS LIST
PROGRAM-II
STUDENT GENERATOR
RE-CODED DATA (CORRECTED)
STUDENT RECORDS
STUDENT RECORDS
PROGRAM-III
M.E.T.E.R. FLOW
PARAMETERS AND RESOURCE DATA
STATISTICS AND REPORTS

Figure B. 1

DATA CONFIGURATION FOR PROGRAM I

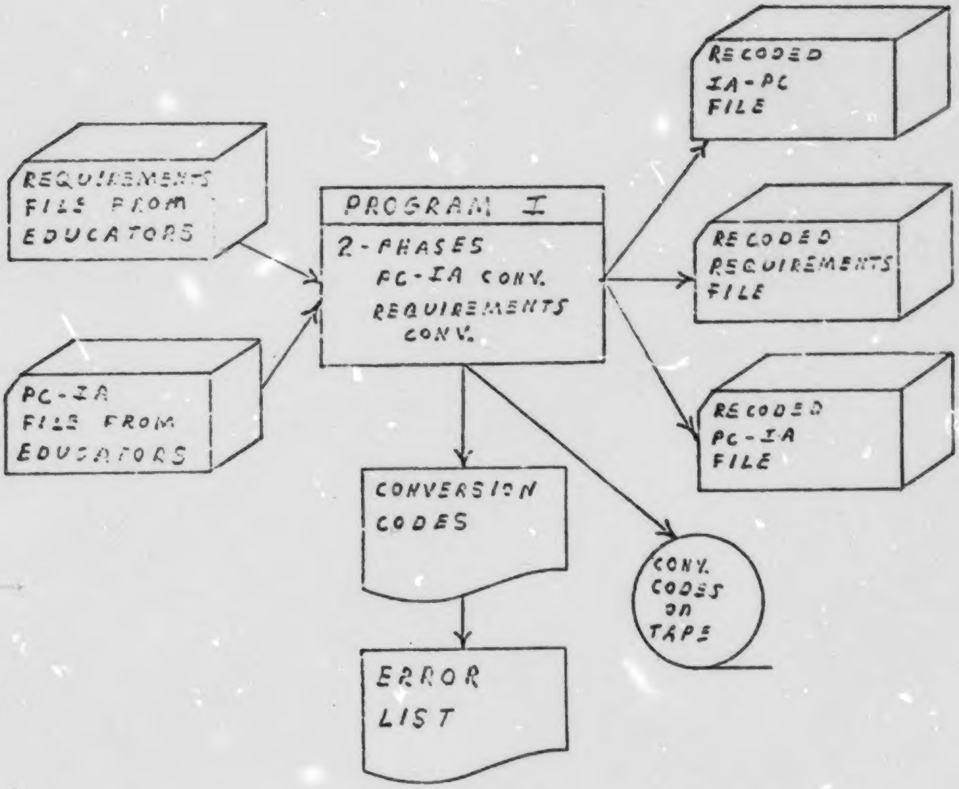

REQUIREMENTS FILE FROM EDUCATORS
PC-IA FILE FROM EDUCATORS
PROGRAM I
2-PHASES
PC-IA CONV.
REQUIREMENTS CONV.
RECODED IA-PC FILE
RECODED REQUIREMENTS FILE
RECODED PC-IA FILE
CONVERSION CODES
ERROR LIST
CONV. CODES on TAPE

Figure B. 2

PHASE 1 OF PROGRAM I - PC-IA CONVERSION

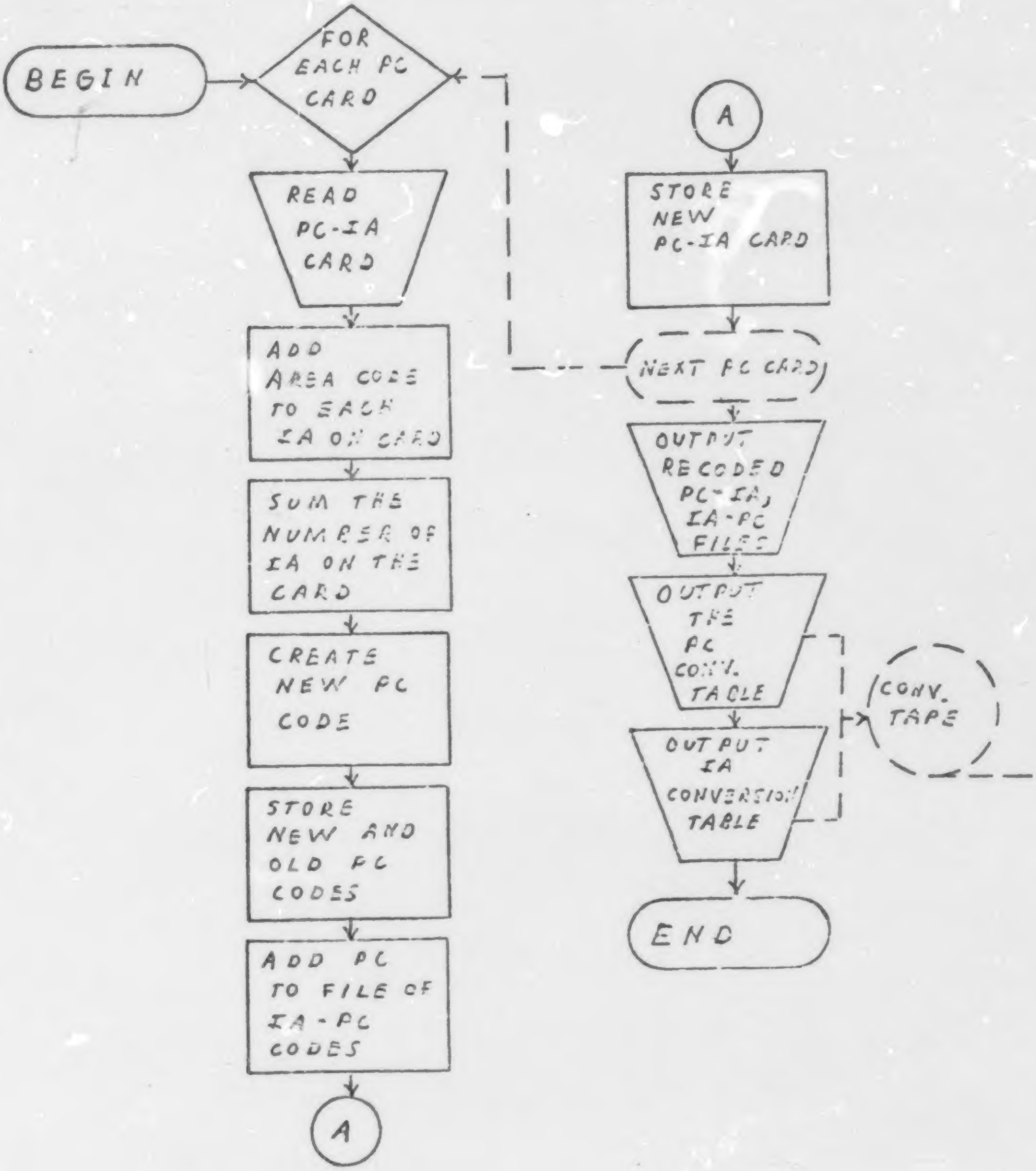

BEGIN
FOR EACH PC CARD
READ PC-IA CARD
ADD AREA CODE TO EACH IA ON CARD
SUM THE NUMBER OF IA ON THE CARD
CREATE NEW PC CODE
STORE NEW AND OLD PC CODES
ADD PC TO FILE OF IA-PC CODES
A
A
STORE NEW PC-IA CARD
NEXT PC CARD
OUTPUT RECODED PC-IA, IA-PC FILES
OUTPUT THE PC CONV. TABLE
CONV. TAPE
OUTPUT IA CONVERSION TABLE
END

Figure B. 3

PHASE 2 OF PROGRAM I - REQUIREMENTS CONVERSION

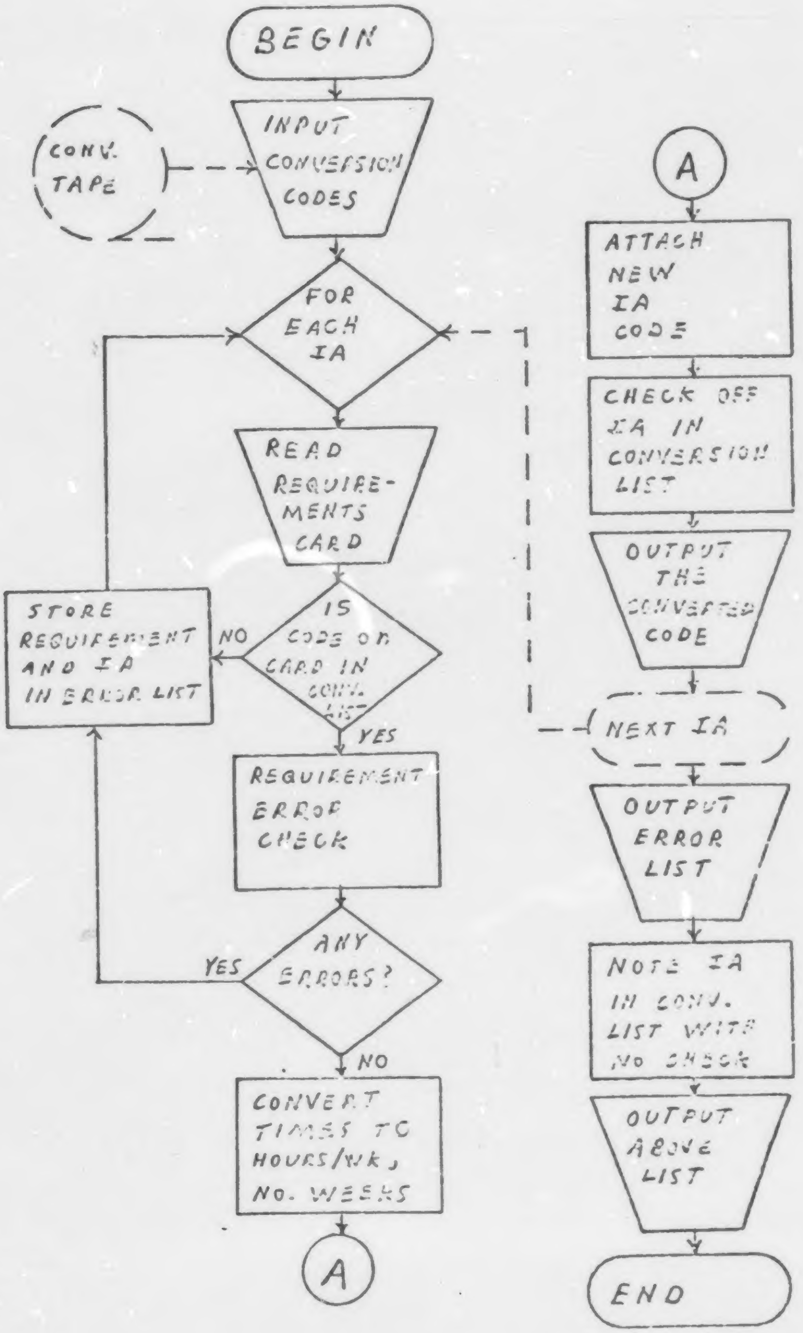

BEGIN
CONV. TAPE
INPUT CONVERSION CODES
FOR EACH IA
READ REQUIREMENTS CARD
IS CODE ON CARD IN CONV. LIST
NO
STORE REQUIREMENT AND IA IN ERROR LIST
YES
REQUIREMENT ERROR CHECK
ANY ERRORS?
YES
NO
CONVERT TIMES TO HOURS/WK, NO. WEEKS
A
A
ATTACH NEW IA CODE
CHECK OFF IA IN CONVERSION LIST
OUTPUT THE CONVERTED CODE
NEXT IA
OUTPUT ERROR LIST
NOTE IA IN CONV. LIST WITH NO CHECK
OUTPUT ABOVE LIST
END

Figure C

DATA CONFIGURATION FOR PROGRAM II

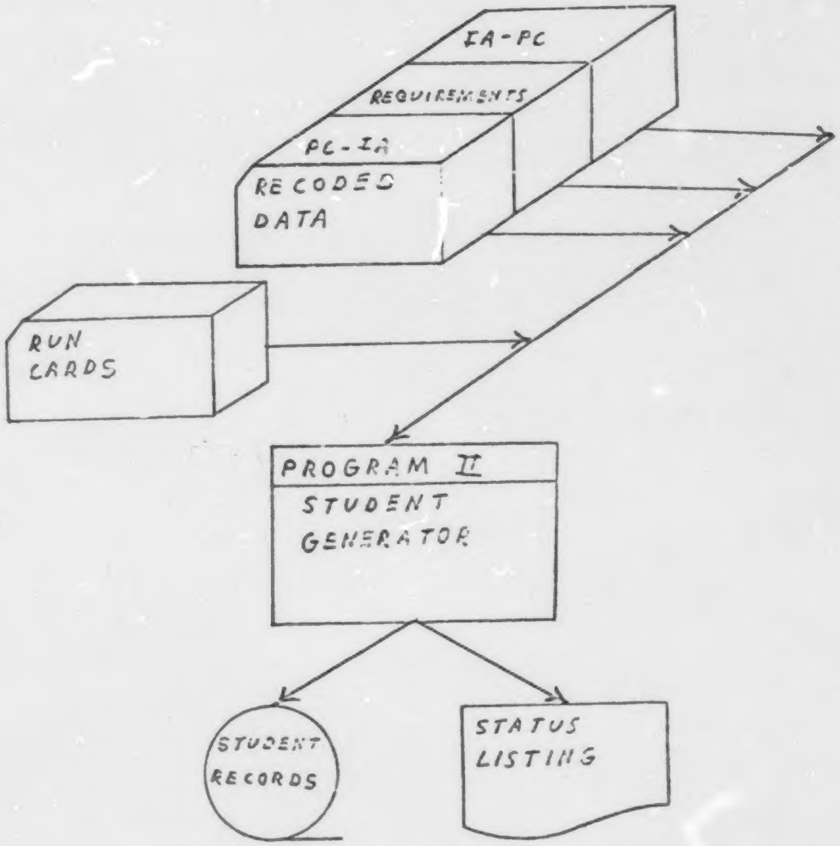

IA-PC
REQUIREMENTS
PC-IA
RECODED DATA
RUN CARDS
PROGRAM II
STUDENT GENERATOR
STUDENT RECORDS
STATUS LISTING

Figure D

DETAIL OF PROGRAM II RUN CARDS

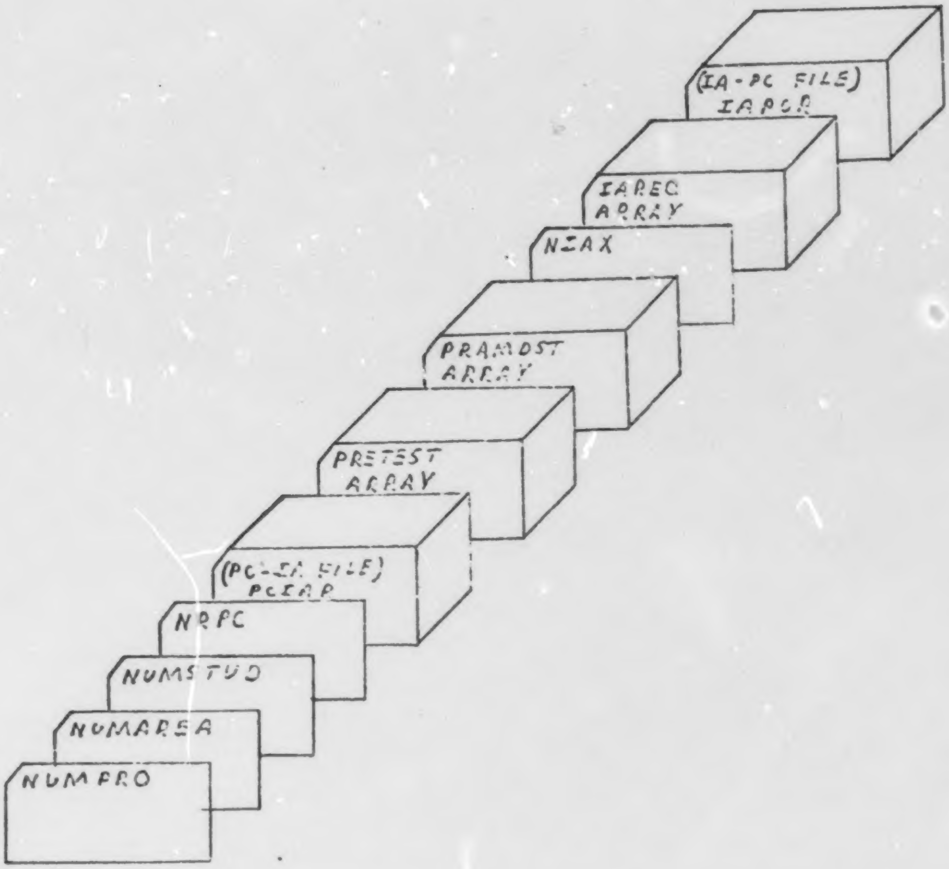

(IA-PC FILE)
IAPCR
IAREC
ARRAY
NIAX
PRAMDST
ARRAY
PRETEST
ARRAY
(PC-IA FILE)
PCIAR
NRPC
NUMSTUD
NUMAREA
NUMPRO

Figure E

FLOWCHART FOR PROGRAM II

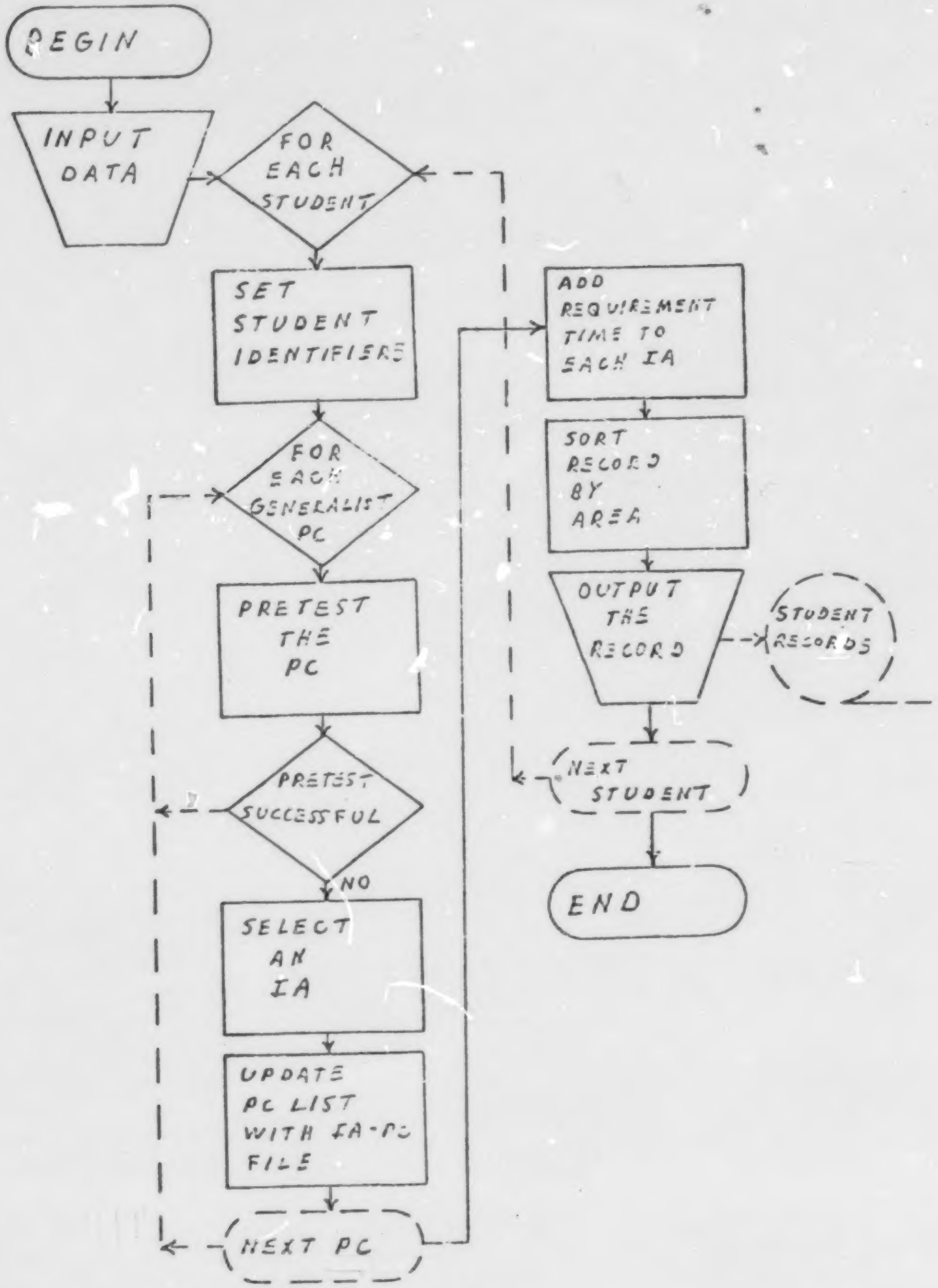

BEGIN
INPUT DATA
FOR EACH STUDENT
SET STUDENT IDENTIFIERS
FOR EACH GENERALIST PC
PRETEST THE PC
PRETEST SUCCESSFUL
Y
NO
SELECT AN IA
UPDATE PC LIST WITH IA-PC FILE
NEXT PC
ADD REQUIREMENT TIME TO EACH IA
SORT RECORD BY AREA
OUTPUT THE RECORD
STUDENT RECORDS
NEXT STUDENT
END

FIGURE F

TABLE OF PROGRAM III SUBPROGRAM NAMES

PROGRAM NAME	CONTENTS
ANARCHY	main program, sets up semester and weekly cycles, inputs students, makes student selections, updates student records, writes weekly statistics
DATAREAD	inputs all the data in a run except the student records, referenced once each run
MATCH 1	allocates resources to student IA requests, indicates codes for no allocation, records resources used in cycle, prints a portion of the semester statistics referenced once each week and in handling semester length requests
MATCH 2	posts allocation results from demand list as the student active list (the CURRNT vector), called once each week and for handling semester length requests
CURPACK	removes any unscheduled IA from students active list, removes record of any graduated student, "packs" the list, calledonce each week and for handling semester length requests.
HOURS	selects the hours per week a student will be willing to devote, selection erased on an external distribution, called once for each student entireing system
IVARND	generates a random v.s. over a specified range, used in picking an area for selecting IA to be demanded, called each time a new area must be chosen for IA requests
RNOG	a random number generator, gives random number between 0 and 1, called as needed.

Figure G

PROGRAM III - BASIC FLOWCHART

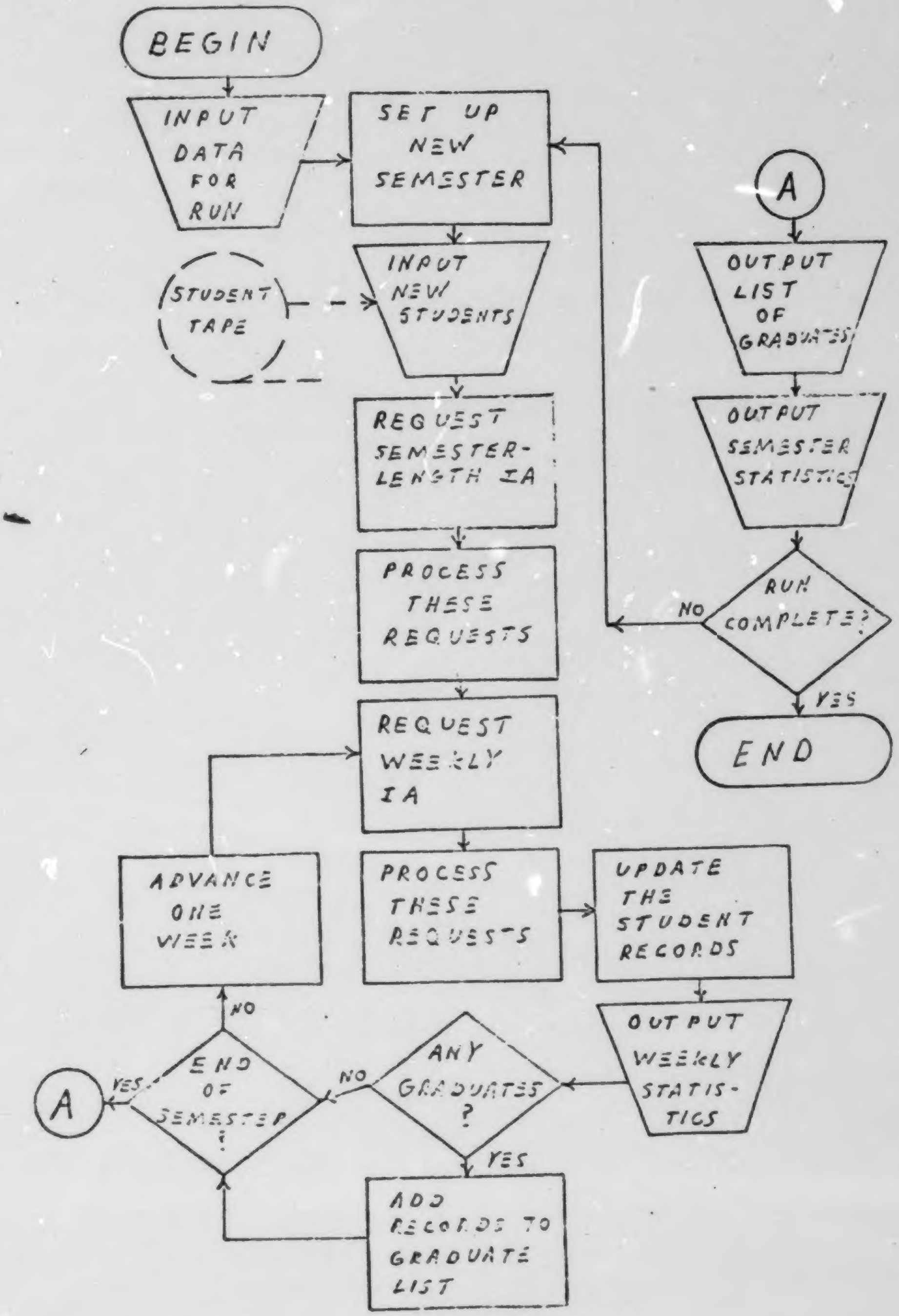

BEGIN
INPUT DATA FOR RUN
SET UP NEW SEMESTER
STUDENT TAPE
INPUT NEW STUDENTS
REQUEST SEMESTER-LENGTH IA
PROCESS THESE REQUESTS
REQUEST WEEKLY IA
ADVANCE ONE WEEK
PROCESS THESE REQUESTS
UPDATE THE STUDENT RECORDS
OUTPUT WEEKLY STATISTICS
ANY GRADUATES ?
NO
YES
ADD RECORDS TO GRADUATE LIST
END OF SEMESTER ?
NO
YES
A
A
OUTPUT LIST OF GRADUATES
OUTPUT SEMESTER STATISTICS
RUN COMPLETE?
NO
YES
END

FIGURE H

TABLE OF INPUT CARDS FOR PROGRAM 3

Unless specified, all numbers are right justified in the fields with no decimal point

CARD	CONTENTS
RE-START	Allows the program to begin in the middle of a run by calling in the tape of the previous semester COL.-5 = 0 IF NO RE-START = one less than the starting semester if a re-start COL.-10= 0 IF NO RE-START = 1 IF RE-START
NUMSEM	number of semesters the program will simulate COLS.1-4=number of semesters
NUMSTUD	array holding the number of students to enter the system each semester COLS 1-4=number entering first semester COLS 5-8=number entering second semester COLS 9-12=number entering third semester and so on in fields of four columns
SCHST	factor used in over-filling the demand list to insure a level of I. A. scheduled. The factor is currently unused but must be given a value (in current runs 1.0) COLS 1-4=number with decimal point
SCHLOW	minimum time in hours a student must have available in any week to attempt demanding further IA COLS 1-4 value of SCHLOW, without decimal.
SLOP	sets the limits about the exact student hours per week that will be deemed acceptable for demanding IA COLS 1-4 value of SLOP, no decimal
STMDST	array of probabilities assigned to the four possible student hours-per-week values the probabilities are CUMULATIVE COLS 1-4 = probability of 20 hours/week 5-8 = probability of 25 hours/week 9-12= probability of 30 hours/week 13-16= probability of 35 hours/week all values have decimal point

Figure H (con't)

TMAX	that per cent of total resources which can be committed to semester length IA COLS 1-4 = value of TMX with decimal point
NIA	total number of IA in requirements file COLS 1-4 = values of NIA, no decimal
NUMRES	total types or classes of physical resources available in this run. COLS 1-4 = values of NUMRES, no decimal
ICRESCAP	array of maximum capacities of each type of physical resource, measured in seats, or stations available COLS 1-4 = capacity of first resource COLS 5-8 = capacity of second resource and so on in groups of four columns
IAREQ	array of requirements, one requirement set per card. Detailed explanation of formats is found in the text of the first part of the appendix. COLS 1-6 - IA code 15-16- hours/week 17-18- number of weeks 25-26- min. no. of students 27-29- Max. no. of students 31 - first faculty type 32-33- hours this type must commit 34 - second faculty type 35-36- hours this type must commit 41-42- physical resource code
NOWEEK	number of weeks in the semester, usually 14 COLS 1-4 = value of NOWEEK
ICRES	array holding the amount of each physical resource available each week, measured in seat - (or station) hours per week. One set of 14 week figures per card, hence, one card per physical resource. COLS. 1-4 = first weeks availability COLS 5-8 = second weeks availability and so on in groups of four columns
NUMINST	number of distinct faculty types COLS 1-4 = value of NUMINST
NUMAREA	number of curriculum areas COLS. 1-4 = value of NUMAREA

Figure H (con't)

NTEACH	array holding the available hours of each faculty type in each area for each week. Each card has 14 values, one per week, and there is one card in each type in each area COLS. 1-4 = hours available in first week COLS. 5-8 = hours available in second week and so on in groups of four columns
IWDRES	physical resource titles, one title for each type. One title per card. COLS 1-16 = any alphanumeric title
IWDAREA	area titles, one for each area COLS 1-16 = any alphanumeric title
IWDINS	faculty type titles, one title on a card COLS 1-16 = any alphanumeric title
IWDIS	titles for disappointment categories. five categories, one title per card COLS 1-16 = any alphanumeric title

Figure I

STRUCTURE OF INPUT DECK FOR PROGRAM III

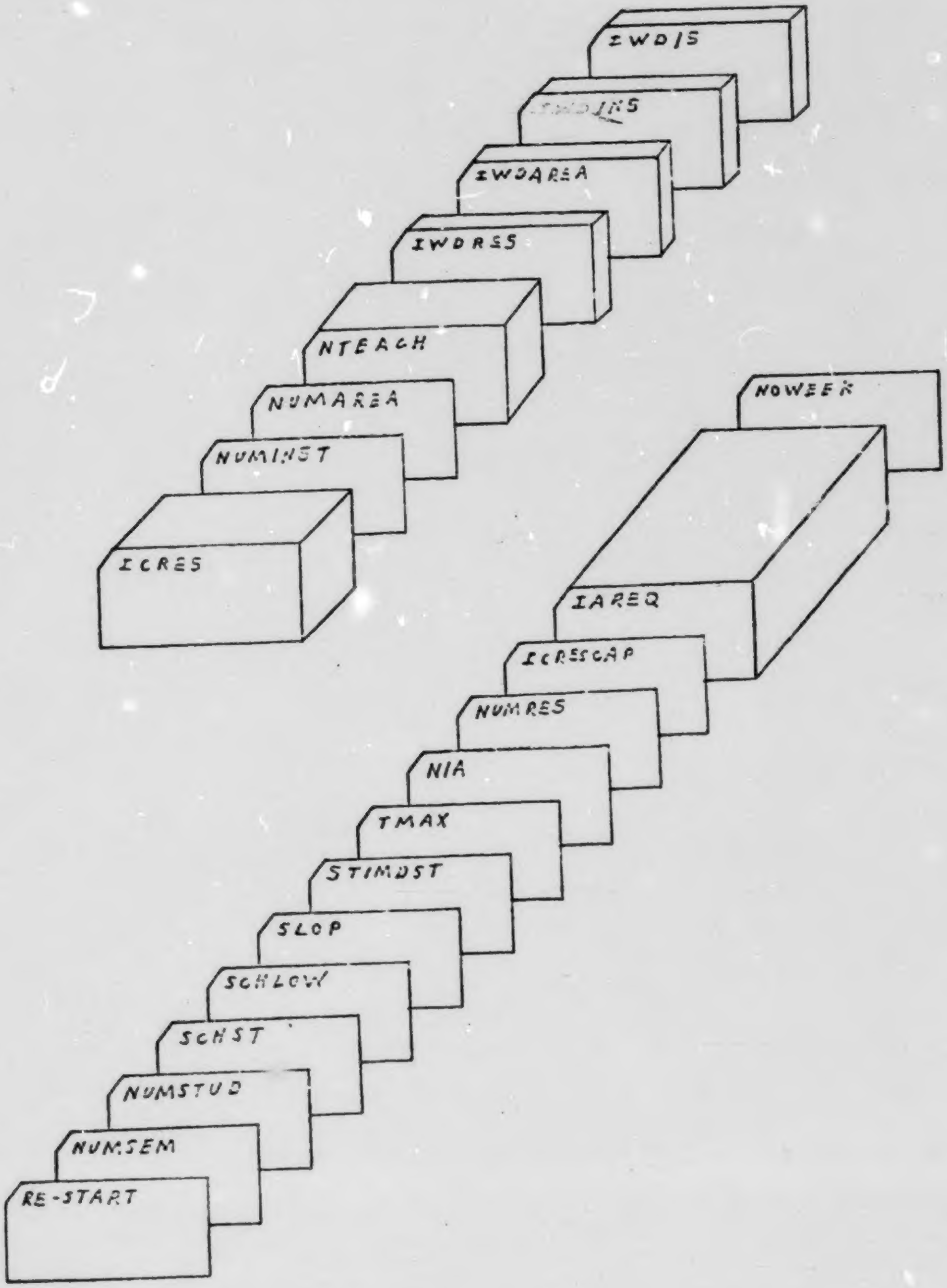

IWDIS
IWDINS
IWDAREA
IWDRES
NTEACH
NUMAREA
NUMINST
ICRES
NOWEEK
IAREQ
ICRESCAP
NUMRES
NIA
TMAX
STIMDST
SLOP
SCHLOW
SCHST
NUMSTUD
NUMSEM
RE-START

Figure J.1

PROGRAM III - ANARCHY - PAGE 1

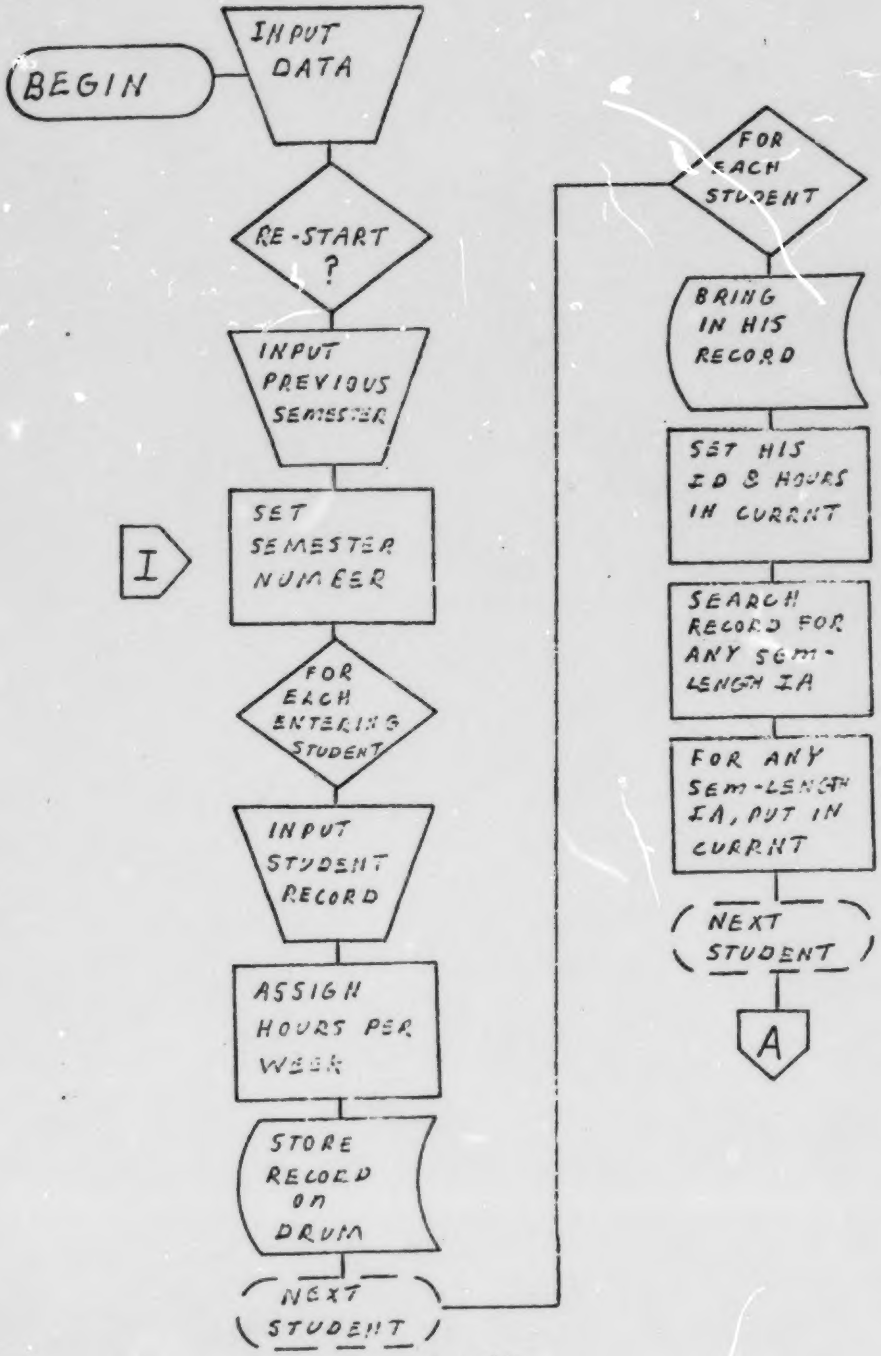

BEGIN
INPUT DATA
RE-START ?
INPUT PREVIOUS SEMESTER
I
SET SEMESTER NUMBER
FOR EACH ENTERING STUDENT
INPUT STUDENT RECORD
ASSIGN HOURS PER WEEK
STORE RECORD ON DRUM
NEXT STUDENT
FOR EACH STUDENT
BRING IN HIS RECORD
SET HIS ID & HOURS IN CURRNT
SEARCH RECORD FOR ANY SEM-LENGTH IA
FOR ANY SEM-LENGTH IA, PUT IN CURRNT
NEXT STUDENT
A

Figure J. 2

PROGRAM III - ANARCHY - PAGE 2

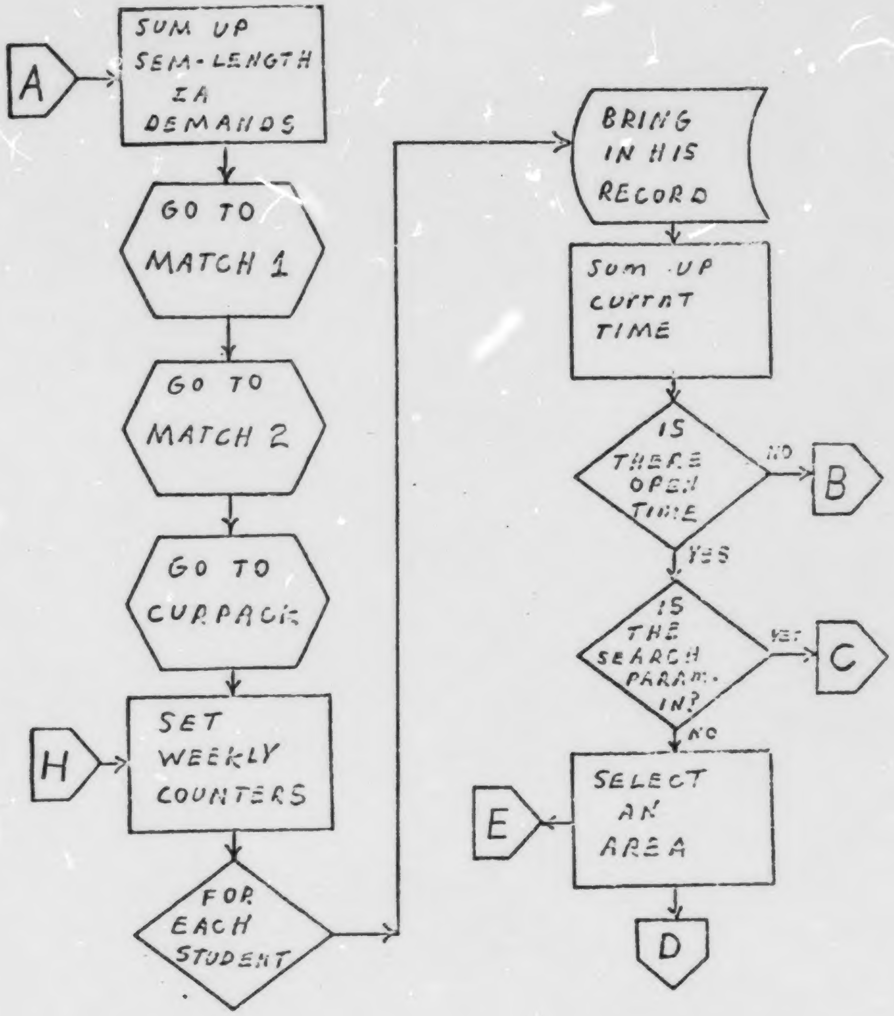

A
SUM UP SEM-LENGTH IA DEMANDS
GO TO MATCH 1
GO TO MATCH 2
GO TO CURPACK
H
SET WEEKLY COUNTERS
FOR EACH STUDENT
BRING IN HIS RECORD
Sum UP currnt TIME
IS THERE OPEN TIME
NO
B
YES
IS THE SEARCH PARAM. IN?
YES
C
NO
SELECT AN AREA
E
D

Figure J. 3

PROGRAM III - ANARCHY - PAGE 3

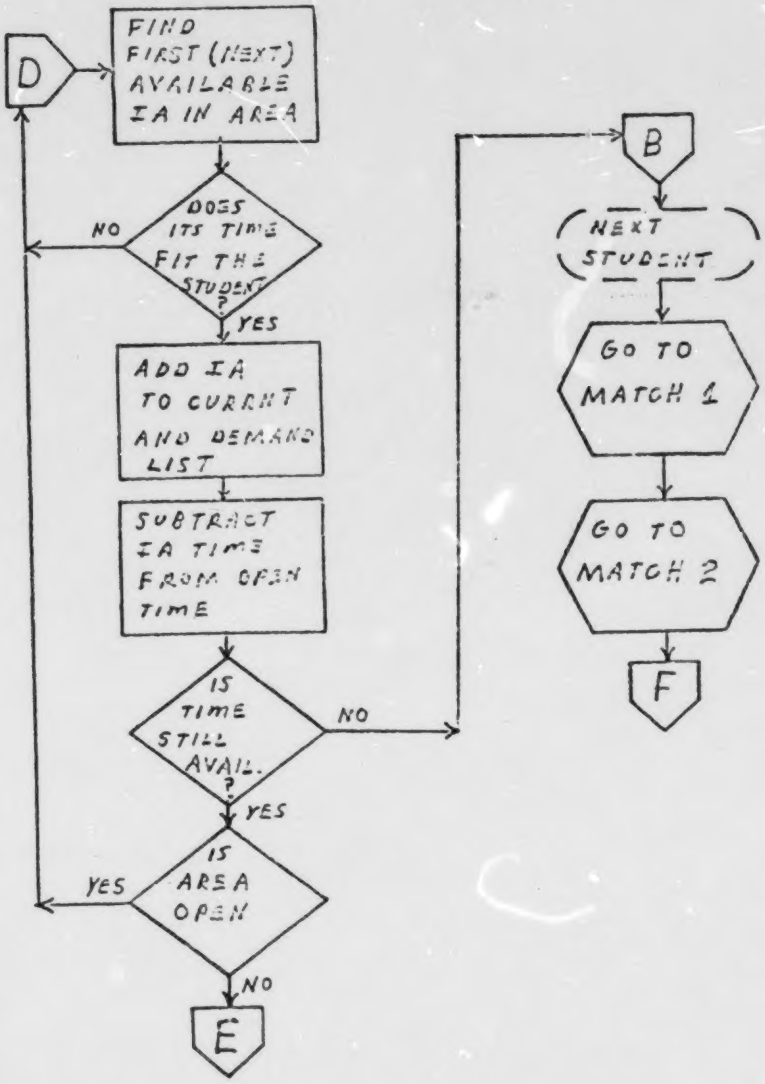

D
FIND FIRST (NEXT) AVAILABLE IA IN AREA
DOES ITS TIME FIT THE STUDENT?
NO
YES
ADD IA TO CURRNT AND DEMAND LIST
SUBTRACT IA TIME FROM OPEN TIME
IS TIME STILL AVAIL.?
NO
YES
IS AREA OPEN
YES
NO
E
B
NEXT STUDENT
GO TO MATCH 1
GO TO MATCH 2
F

Figure J. 4

PROGRAM III - ANARCHY - PAGE 4

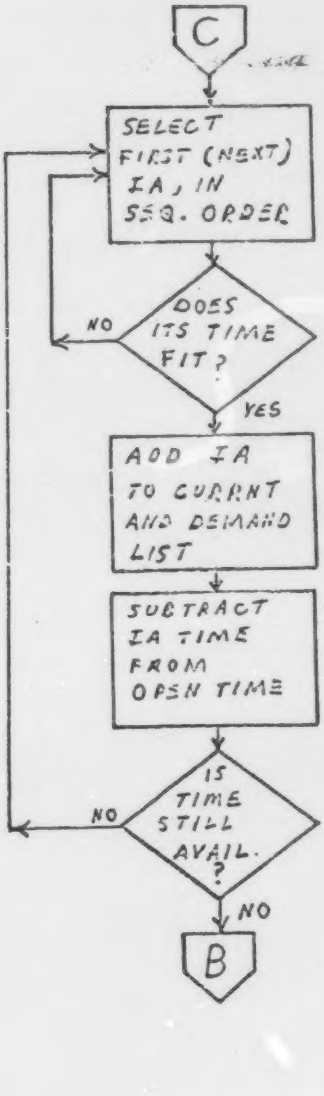

C
SELECT FIRST (NEXT) IA, IN SEQ. ORDER
DOES ITS TIME FIT?
NO
YES
ADD IA TO CURRNT AND DEMAND LIST
SUBTRACT IA TIME FROM OPEN TIME
IS TIME STILL AVAIL. ?
NO
NO
B

Figure J. 5

PROGRAM III - ANARCHY - PAGE 5

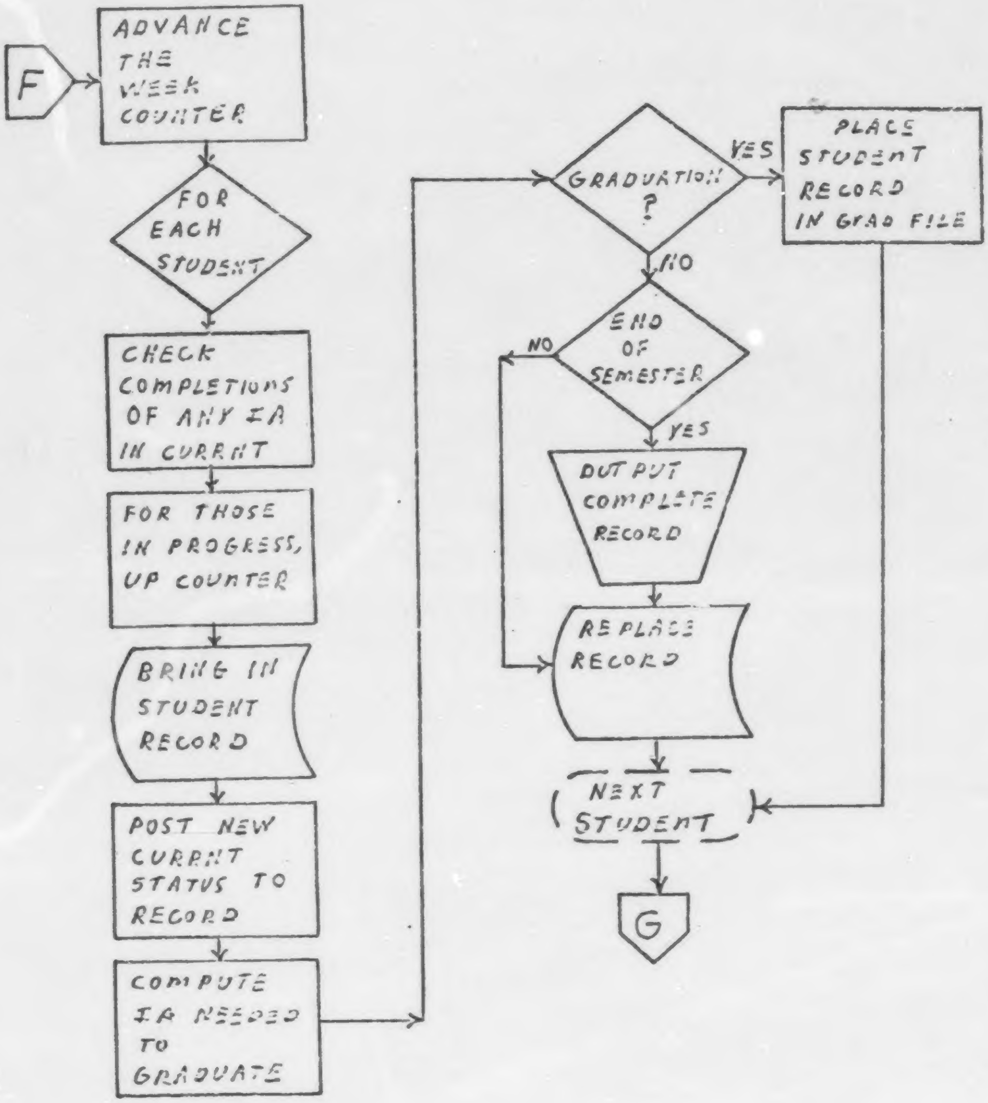

F
ADVANCE THE WEEK COUNTER
FOR EACH STUDENT
CHECK COMPLETIONS OF ANY IA IN CURRNT
FOR THOSE IN PROGRESS, UP COUNTER
BRING IN STUDENT RECORD
POST NEW CURRNT STATUS TO RECORD
COMPUTE IA NEEDED TO GRADUATE
GRADUATION ?
YES
PLACE STUDENT RECORD IN GRAD FILE
NO
END OF SEMESTER
NO
YES
OUTPUT COMPLETE RECORD
REPLACE RECORD
NEXT STUDENT
G

Figure J. 6

PROGRAM III - ANARCHY - PAGE 6

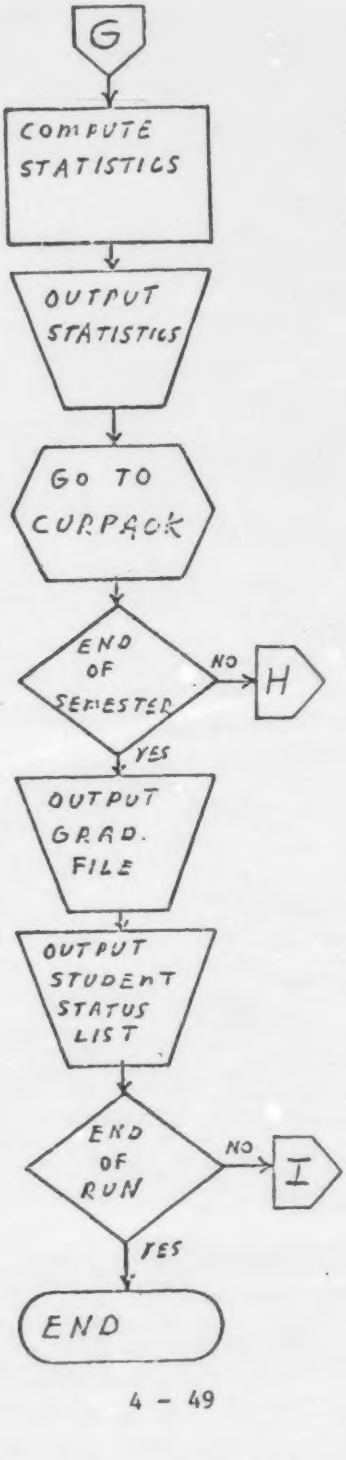

G
COMPUTE STATISTICS
OUTPUT STATISTICS
GO TO CURPACK
END OF SEMESTER
NO
H
YES
OUTPUT GRAD. FILE
OUTPUT STUDENT STATUS LIST
END OF RUN
NO
I
YES
END

Figure J. 7

PROGRAM III - SUBROUTINE MATCH 1 - PAGE 1

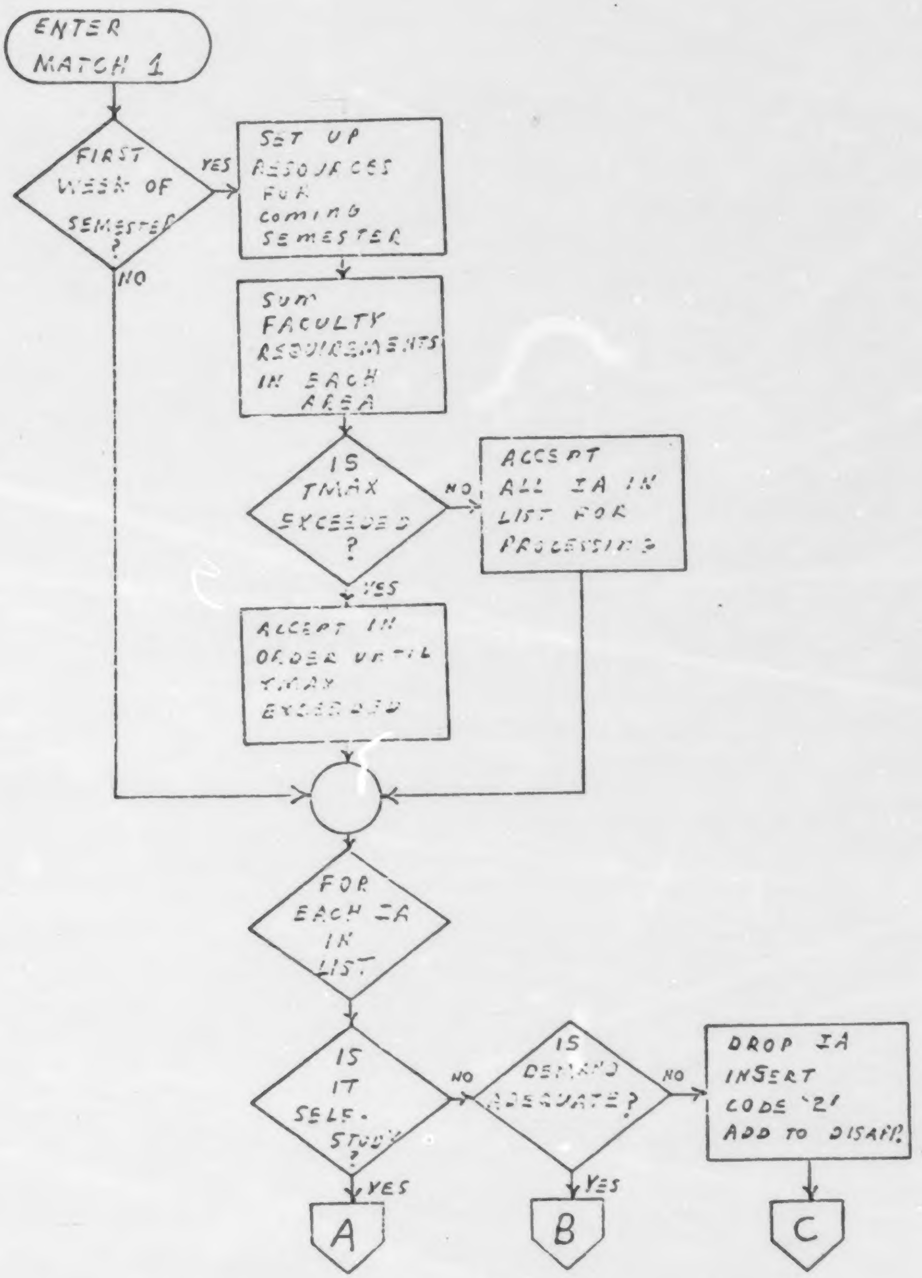

ENTER MATCH 1
FIRST WEEK OF SEMESTER ?
YES
NO
SET UP RESOURCES FOR COMING SEMESTER
SUM FACULTY REQUIREMENTS IN EACH AREA
IS TMAX EXCEEDED ?
NO
YES
ACCEPT ALL IA IN LIST FOR PROCESSING
ACCEPT IN ORDER UNTIL TMAX EXCEEDED
FOR EACH IA IN LIST
IS IT SELF-STUDY ?
NO
YES
IS DEMAND ADEQUATE?
NO
YES
DROP IA INSERT CODE '2' ADD TO DISAPP.
A
B
C

Figure J. 8

PROGRAM III - SUBROUTINE MATCH-1 - PAGE 2

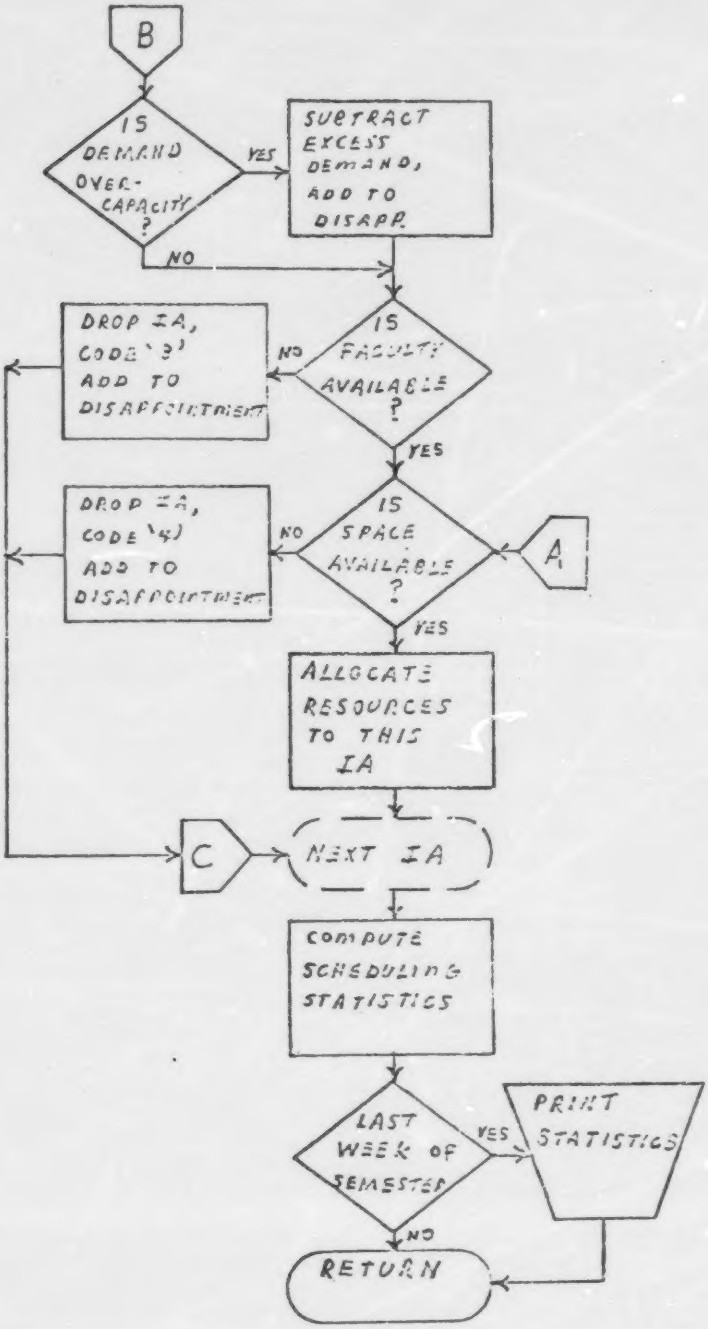

B
IS DEMAND OVER-CAPACITY ?
YES
SUBTRACT EXCESS DEMAND, ADD TO DISAPP.
NO
IS FACULTY AVAILABLE ?
NO
DROP IA, CODE '3' ADD TO DISAPPOINTMENT
YES
IS SPACE AVAILABLE ?
A
NO
DROP IA, CODE '4' ADD TO DISAPPOINTMENT
YES
ALLOCATE RESOURCES TO THIS IA
C
NEXT IA
COMPUTE SCHEDULING STATISTICS
LAST WEEK OF SEMESTER
YES
PRINT STATISTICS
NO
RETURN

Figure J. 9

PROGRAM III - SUBROUTINE MATCH 2

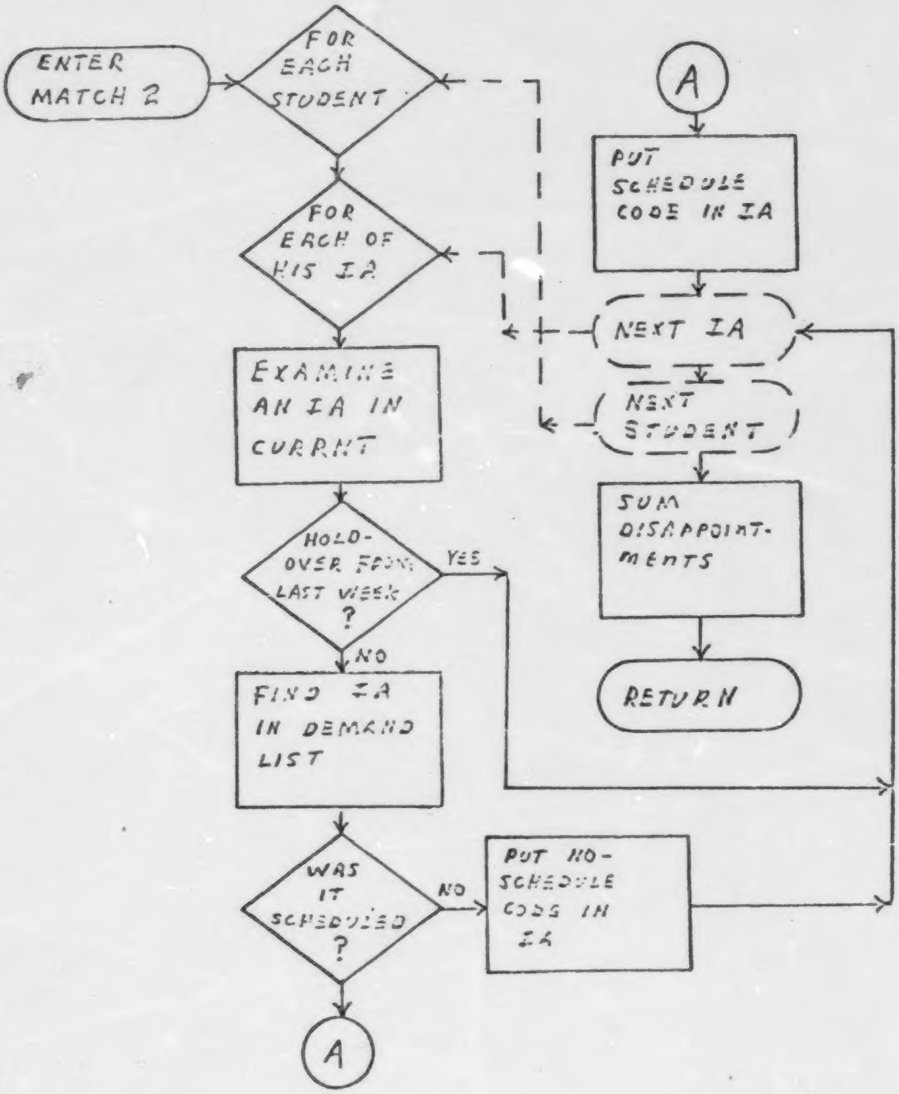
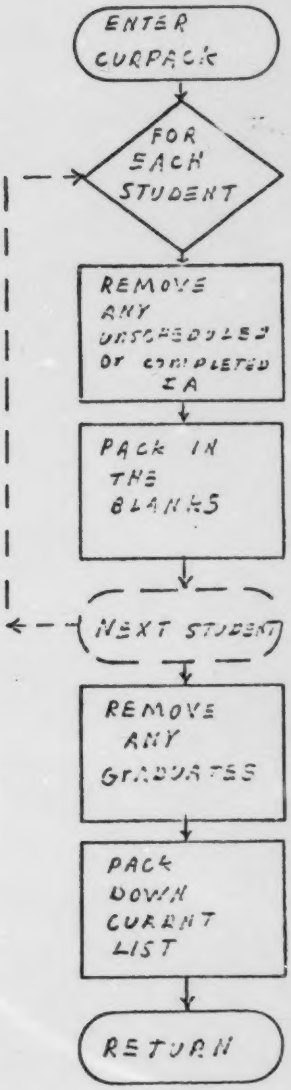

ENTER MATCH 2
FOR EACH STUDENT
FOR EACH OF HIS IA
EXAMINE AN IA IN CURRNT
HOLD-OVER FROM LAST WEEK ?
YES
NO
FIND IA IN DEMAND LIST
WAS IT SCHEDULED ?
NO
PUT NO-SCHEDULE CODE IN IA
A
A
PUT SCHEDULE CODE IN IA
NEXT IA
NEXT STUDENT
SUM DISAPPOINTMENTS
RETURN

Figure J. 10

PROGRAM III - SUBROUTINE CURPACK


ENTER CURPACK
FOR EACH STUDENT
REMOVE ANY UNSCHEDULED OR COMPLETED IA
PACK IN THE BLANKS
NEXT STUDENT
REMOVE ANY GRADUATES
PACK DOWN CURRNT LIST
RETURN

```
DEMAND,24000
EQUIP,3=(STUDENT-FILE),DA,MT
FTN,L,X
      PROGRAM STUDENT

  THIS PHASE OF THE EDSIM PROGRAM CREATES STUDENTS, GIVES THEM
  ELECTIVE PC ASSIGNMENTS, PRETESTS ALL PC, AND SELECTS IA FORTHOSE
  PC NOT PRETESTED.
  THE PROGRAM WILL CREATE ANY NUMBER OF STUDENTS FOR 14 OR LESS PROFILES
  WRITTEN BY WILLIAMS AND LEININGER - 7/18/69

        RUN OF AUGUST 16  -  NOTE CHANGES IN COMMON /A/
      ALSO NOTE THAT ONLY 4 PC ACROSS ARE READ IN THE PC/IA FILE
      LIKEWISE FOR THE IA/PC FILE
      THE IAREQ FILE HOLDS ONLY THE FIRST 5 SLOTS

      COMMON /1/ PCIAR(5,590), IAPCR( 950,8), IAREQ( 950,5),
     1NPC(11),NUMSTUD(5),PRMDST(5),NUMPRO,NUMAREA,NRPC
      DIMENSION STREC(500), PCIAR2(5,600)
      INTEGER STREC, PCIAR, PCIAR2, PCIAE, PFILE, PRO,PRE
      DUM = RNOG(1)
      CALL DATAREAD
      ID = 0
      DO 12 I=1,5     $     DO 12 J=1,NRPC
   12 PCIAR2(I,J) = PCIAR(I,J)

  MOST OUTER LOOP - DONE FOR EACH PROFILE

  FOR FIRST RUN - ONLY ONE PROFILE
      PRO = 1
  LOOP FOR CREATING THE STUDENTS FOR THIS PROFILE
      NUM = NUMSTUD(PRO)
      DO 500 ISTD=1,NUM
      ISTCT = 0     $     IEPC = 0
      DO 102 J=1,500
  102 STREC(J) = 0
      DO 103 I=1,5     $     DO 103 J=1,NRPC
  103 PCIAR(I,J) = PCIAR2(I,J)
  CREATE FIRST STREC VALUES
      STREC(1) = ID + ISTD
      STREC(3) = PRO
      P R M D S T  R E M O V E D
      STREC(4) = 3

  ENTER THE SECTION FOR ALL STUDENTS-THE GENERALIST PILE
      DO 200 NREQ=1,NRPC
  CHECK EACH PC FOR THE GENERALIST CODE
  114 IAREA = PCIAR(1,NREQ)/100000
      TEST FOR AREA = 77, SAYING PC ARREADY MET
      IF(IAREA .EQ. 77) 120,121
  PUT COUNTER FOR 77S HERE
  120 GO TO 200
      PRETEST THIS PC
  121 IPT = PTEST(IAREA)
      GO TO (124,122) IPT
  SUCCESSFUL PRETEST
```

```
124 IWK = MOD(PCIAR(1,NREQ),100000)
    PCIAR(1,NREQ) = 7700000 + IWK
    GO TO 200
122 IAS = PCIAR(1,NREQ)/10000
    IAS = MOD(IAS,10)
    IF (IAS .EQ. 1) 141,143
141 IASEL = PCIAR(2,NREQ)
    GO TO 144
    TO GET IA POSITION, CALL DIST
143 CALL DIST (IAS,STREC(4),IANUM)
    IASEL = PCIAR(IANUM+1,NREQ)
144 IAXPN = IASEL * 100000000
    ISTCT = ISTCT + 1
    IF (ISTCT .GT. 500) 146,148
146 PRINT 728
    ISTCT = 500
    GO TO 402
148 CONTINUE
    STREC(ISTCT)=IAXPN
    WORK ON OTHER PC
    IADD = MOD(IASEL,10000)
    IF (IASEL .EQ. IAPCR(IADD,1)) 129,127
127 CONTINUE
    PRINT 706, IASEL  $  GO TO 999
129 DO 135 J=2,8
    IF (IAPCR(IADD,J) .EQ. 0) GO TO 135
    DO 132 LP=NREQ,NRPC
    IF (PCIAR(1,LP) .EQ. IAPCR(IADD,J)) 130,132
130 IWK = MOD(PCIAR(1,LP),100000)
    PCIAR(1,LP) = 7700000 + IWK
132 CONTINUE
135 CONTINUE
 FINISHED ONE GENERALIST PC
200 CONTINUE
 DONE WITH ALL AREAS  -  STREC IS NOW COMPLETE
402 CONTINUE
    STREC(6) = ISTCT - 9
 SORT AND WRITE STREC
    UPDATE STREC WITH IAREQ
    DO 420 IST=10,ISTCT
    IASHT = STREC(IST)/100000000
    IADD = MOD(IASHT,10000)
    IF (IASHT .EQ. IAREQ(IADD,1)) 412,410
410 CONTINUE
    ITIME = 1
    NUMWK = 1
    PRINT 718, IASHT
    GO TO 415
412 ITIME = IAREQ(IADD,2)
    NUMWK = IAREQ(IADD,3)
```

000

```
    HANDLE INDEP STUDY-PUT 9 IN D SLOT
    IF (IAREQ(IADD,5) .EQ. 999) 415,417
415 STREC(IST) = STREC(IST) + 9000000+ITIME*10000+NUMWK*100                000
    GO TO 420
417 STREC(IST) = STREC(IST) + ITIME*10000 + NUMWK*100
420 CONTINUE
    STREC SORT BY AREA
    MOVLIM = ISTCT-1
    DO 425 IMOV=10,MOVLIM
    IMOV1 = IMOV + 1
    IHOLD = IMOV
    DO 422 JMOV=IMOV1,ISTCT
    ISTONE = STREC(JMOV)/100000000
    ISTTWO = STREC(IHOLD)/100000000
    IF (ISTONE .LE. ISTTWO) 421,422
421 IHOLD = JMOV
422 CONTINUE
    ITEMP1 = STREC(IMOV)
    STREC(IMOV) = STREC(IHOLD)
    STREC(IHOLD) = ITEMP1
425 CONTINUE
 BUFFER OUT STREC
    BUFFER OUT (3,1) (STREC(1),STREC(500))
430 IF(UNIT,3) 430,432,434,434
432 PRINT 441, STREC(1)
500 CONTINUE
 DONE WITH ALL STUDENTS IN THIS PROFILE
    ID = STREC(1)
 HERE WILL BE LOOP FOR COMPLETION OF ALL PROFILES
    PRINT 443, PRO
    PRINT 445
    REWIND 3
    STOP
999 PRINT 701, (STREC(I),I=1,8)
    STOP
434 PRINT 435    $    STOP
435 FORMAT (/,5X,6H* * * ,*EOF OR PARITY ON LU 3*)
441 FORMAT (10X,*STUDENT *,I4,* HAS BEEN BUFFERED OUT*)
443 FORMAT (5X,*COMPLETION OF PROFILE *,I2)
445 FORMAT (/,5X,*ALL PROFILES COMPLETED - NORMAL TERMINATION*)
701 FORMAT (5X,6H* * * ,*STREC DUMP DUE TO ERROR*,5X,8I10)
702 FORMAT (/,5X,6H* * * ,*ERROR-NO PARAMETER SELECTED-3 PUT IN*,5X,
   1*ID =*,I6,/)
704 FORMAT (/,5X,6H* * * ,*ERROR - NONGENERALIST PC FOUND IN PCIAR*,
   15X,*NREQ =*,I6,/)
706 FORMAT (/,5X,6H* * * ,*ERROR-NO IA IN IAPCR FOR IASEL*,5X,*IASEL =
   1*,I8,/)
710 FORMAT (/,5X,6H* * * ,*ERROR-IX .GT. NUMBER OF OPEN PCIAE*,5X,
   1*IX =*,I4,2X,*AREA =*,I2,/)
712 FORMAT (/,5X,6H* * * ,*ERROR-ELECTIVE IASEL = 0*,5X,*PCIAE =*,
                                                                           00
   1I8,2X,*IANUM =*,I2,/)
714 FORMAT (/,5X,6H* * * ,*ERROR-NO IASEL/IAPCR MATCH*,5X,*IASEL =*,
   1I8,2X,*AREA =*,I2,/)
716 FORMAT (/,5X,6H* * * ,*ERROR-IAPCR 4/D DOSENT MATCH PCIAE4/D*,5X,
   1*AREA =*,I2,2X,*IAPCR 4/D*,I4,2X,*PCIAE 4/D*,I4,/)
718 FORMAT (//,12H* * * * * * ,*NO IAREQ FOR IA *,I8,*  SUBSTISUTION M
   1ADE IN THIS PROGRAM ONLY*,//)
728 FORMAT (10X,*THE FOLLOWING STUDENTS IA LIST EXCEEDED 500*)
    END
    SUBROUTINE DATAREAD
    COMMON /1/ PCIAR(5,590), IAPCR(950,8), IAREQ(950,5),
   1NPC(11),NUMSTUD(5),PRMDST(5),NUMPRO,NUMAREA,NRPC
    COMMON /2/ PRETEST(14)
    COMMON /3/ PRAMDST(5,6,5)
    DIMENSION IAPCRHD(8)
    DIMENSION IREQHLD(5)
    INTEGER PCIAR, PCIAE, PFILE
C   INPUT NUMBER OF PROFILES, MAXIMUM = 14
    READ (60,101) NUMPRO
C   INPUT NUMBER OF AREAS, MAXIMUM = 14
    READ (60, 101) NUMAREA
C   INPUT NUMBER OF ELECTIVE PC EACH AREA,MAXIMUM=100 PER AREA
    READ (60,101) (NPC(I), I=1, NUMAREA)
C   INPUT NUMBER OF STUDENTS TO BE CREATED PER PROFILE
    READ(60,101)(NUMSTUD(I),I=1,NUMPRO)
C   INPUT NUMBER OF ELECTIVES EACH AREA FOR EACH PROFILE
C   INPUT TOTAL NUMBER OF REQUIRED PC MAXIMUM=400
    READ(60,101)NRPC
C   INPUT PCIAR MATRIX
    READ(60,102)((PCIAR(J,I),J=1,5),I=1,NRPC)                         ALLC
C   INPUT PRETEST PROBABILTIES
    READ(60,103)(PRETEST(I),I=1,NUMAREA)
C   INPUT PRAMDST PROBABILITY DISTRIBUTION
    DO 20 J=1,5
    DO 20 K=1,5
 20 READ(60,103)(PRAMDST(K,L,J),L=1,6)
    READ(60,101)NIAX
    DO 3 I=1,NIAX $  READ 104,(IREQHLD(J),J=1,5)
    IADD=MOD(IREQHLD(1),10000)  $  DO 7K=1,5
  7 IAREQ(IADD,K)=IREQHLD(K)
  3 CONTINUE
    DO 30 I=1,NIAX $  READ 105,(IAPCRHD(J),J=1,8)
    IADD=MOD(IAPCRHD(1),10000)  $  DO 25 K=1,8
 25 IAPCR(IADD,K)=IAPCRHD(K)
 30 CONTINUE
C   INPUT FORMAT
101 FORMAT(14I4)
102 FORMAT (I7,3X,4(I6,4X))                                           ALLC
103 FORMAT(14F4.0)
104 FORMAT(I6,8X,2I2,6X,I2,I3)                                        ALLC
                                                                         0
```

```
5 FORMAT (I6,4X,8(I7,1X))
  RETURN      $     END
  SUBROUTINE DIST (IAIN,PARAM,IAOUT)
  COMMON /3/ PRAMDST(5,6,5)
  DIMENSION HOLD(6)
  INTEGER PARAM
  MOVE CURRENT PARAMETER-BASED IA DIST INTO HOLD
  DO 15 I=1,6
5 HOLD(I)=PRAMDST(IAIN-1,I,PARAM)
  ARAND = RNOG(2)
  ARAND = ARAND/100.
  IF(ARAND.EQ.0.99)ARAND=.98
  DO 20 ISAVE=1,IAIN
0 CONTINUE
  PRINT 55, ARAND
  PRINT 50 ,IAIN, (HOLD(J),J=1,6)
  IF (ARAND - HOLD(ISAVE)) 25,25,20
  FORCE FEEDING ISAVE=1 IF FAILURE ON TEST
  ISAVE = 1
  CALL AN ERROR-ROUTINE STOPPING THE PROGRAM
5 IAOUT = ISAVE
0 FORMAT(//,5X,*ERROR IN DIST -NO IAOUT FOUND*,5X,I3,6F10.5//)
5 FORMAT (5X,*RANDOM NUMBER =*,F6.3)
  RETURN          $      END
  FUNCTION IPC9(IPCIA)
  DROPS A 9 IN THE THIRD PC SLOT
  IDD = MOD(IPCIA,10000)
  IAR = IPCIA/100000
  IPC9 = IAR*100000 + 90000 + IDD
  RETURN     $     END
  FUNCTION PTEST(IA)
  COMMON /2/ PRETEST(14)
  PERFORMS THE PRETEST FOR A PC
     PTEST=1 IS PASS          PTEST=2 IS FAIL
  ALOW = PRETEST(IA) - .01
  RAND = RNOG(2)
  RAND = RAND/100.
  IF (RAND .LE. ALOW) 8,10
  PTEST = 1.
  RETURN
  PTEST = 2.
  RETURN     $     END
  FUNCTION IVARND(NSLOTS)
  FUNCTION TO GIVE A RANDOM NUMBER BETWEEN 1 AND THE INPUTED
  VALUE OS NSLOTS
  IF (NSLOTS .LE. 1) 8,10
  IVARND = 1
  RETURN
  SLOT = NSLOTS
  DIV = 1./SLOT
  RAND = RNOG(7)
  HOLD = RAND/DIV
  IVA = HOLD+1.
  IF (IVA .GT. NSLOTS) 15,20
```

```
5 PRINT 30, NSLOTS, IVA
   IVA = NSLOTS
0 IVARND = IVA
0 FORMAT (//,5X,*ERROR IN IVARND - VALUE RETURNED IS NOT PROPER*,
 15X,*NSLOTS =*,I5,5X,*IVA =*,I5)
  RETURN     $     END
  FUNCTION RNOG(ICODE)
  REVISED 5/11 WITH EXPANDED CODE
     ICODE = 1               SET SEED
     ICODE = 2 THRU 5        RETURN INTEGER, LENGTH OF ICODE
     ICODE = 7               RETURN REAL 0 TO 1, LENGTH 5 DIGITS
  IF (ICODE .GT. 1) GO TO 15
  IA = 998877
5 IFAC = IA * 455470314
  IB = MOD(IFAC,214783646)
  IC = IABS(IB)/100
  NOW SPLIT VIA CODE
  IF (ICODE .EQ. 1) 17,19
7 INDEX = 100     $     GO TO 25
9 IF (ICODE .EQ. 7) GO TO 30
  INDEX = 10**ICODE
5 RNOG = MOD(IC,INDEX)
  GO TO 35
0 R = MOD(IC,100000)
  RNOG = R/100000
5 IA = IB
  RETURN  $  END
```

```
,10
ND,24000
P,3=(CODE-CONVERT),MT,DA
L,X
 PROGRAM COMPACT

 COMMON /1/ IAPC(1500,10), IACODE(4), IGEOPC(5)

 THIS PROGRAM IN THE EDSIM SERIES CONVERTS THE PC AND IA CODES
 TO A FORM USEABLE TO THE EDSIM STUDENT GENERATOR.
  WRITTEN BY LEININGER AND WILLIAMS  -  AUGUST 69.

 INTERNAL LOGICAL UNITS ARE 7 AND 9
 LU 3 = OUTPUT TAPE CONVERSION FILE

 FOR THE RUN OF 8/14 - PUNCH (62) HAS BEEN EQUIPED TO 6
 FOR THE RUN OF 8/14 - PUNCH (62) HAS BEEN EQUIPED TO LU 4
 LU 4 = OUTPUT TAPE OF PC-IA AND IA-PC FILES

 DO 3 J=1,1500     $     DO 2 I=2,10
 IAPC(J,I) = 0
 IAPC(J,1) = 1
 KOUNT = 0     $     KK=0
 READ 101, IPC, (IACODE(I),I=1,4)
 IF (EOF,60) 30,13
 CONTINUE

 ADDITION FOR REVISED DATA OF 10/28
 DROPS THE AREA OF THE PC ON TOP OF EACH IA
 IAR = IPC/1000000
 IAR = IAR*10000
 DO 9 K=1,4
 IF (IACODE(K) .EQ. 0) GO TO 9
 IACODE(K) = IACODE(K) + IAR
 CONTINUE

 DO 8 K=1,5
 IGEOPC(K) = 0
 ISUM = 0
 COUNT UP THE IA
 DO 15 J=1,4
 IF (IACODE(J) .EQ. 0) 15,17
 ISUM = ISUM+1
 CONTINUE
 MOVE AREA, G/S, AND NUMBER OF IA
 IAREA = IPC/1000000
 IGS = IPC/10
 IGS = MOD(IGS,10)
 IPACK = IAREA*100000 + ISUM*10000 + IGS*1000
 MOVE IN SEQUENTIAL 3 DIGIT CODE
 KOUNT = KOUNT + 1
 I3D = KOUNT
 IPACK = IPACK + I3D
 LU 7 HAS THE PC CONVERSION FILE
```

```
  WRITE (7,103) IPC, IPACK
  IGEOPC(1) = IPACK
  FLIP AND STORE THE IA-PC FILE
  DO 70 I=1,ISUM    $    DO 71 J=1,KK
  IF (IACODE(I) .EQ. IAPC(J,2)) 75,71
5 IAINDX = IAPC(J,1) + 1
  IF (IAINDX .LE. 10) 22,24
4 PRINT 141, IACODE(I), (IAPC(J,L),L=1,10)
1 FORMAT (10X,*I A I N D X  O V E R F L O W*,2X,I6,10(5X,I8))
  IAINDX = 10
2 CONTINUE
  IAPC(J,1) = IAPC(J,1) + 1
  IAPC(J,IAINDX) = IPACK
  IGEOPC(I+1) = (IAREA*10000) + J
  GO TO 70
1 CONTINUE
  KK = KK + 1
  IAPC(KK,1) = 3
  IAPC(KK,2) = IACODE(I)
  IAPC(KK,3) = IPACK
  IGEOPC(I+1) = IAREA*10000 + KK
0 CONTINUE
  LU 9 HAS NEW PC-IA FILE
  WRITE (9,109) (IGEOPC(J),J=1,5)
  GO TO 77
  FROM EOF, REWIND AND PRINT CONVERSION FILE
0 REWIND 7  $    REWIND 9
   OUTPUT THE PC-IA FILE

  PRINT 120
  DO 32 I=1,1000
  READ (9,109) (IGEOPC(J),J=1,5)
  IF (EOF,9) 33,31
1 PRINT 107, (IGEOPC(J),J=1,5)
  WRITE(62,127) (IGEOPC(J),J=1,5)
2 CONTINUE
3 CONTINUE
  FOR THE TAPE OUTPUT OF THE PUNCH FILE                                    TAPE-4
  IGEOPC(1) = 9999999                                                      TAPE-4
  DO 88 J=2,5                                                              TAPE-4
8 IGEOPC(J) = 0                                                            TAPE-4
  WRITE(62,127) (IGEOPC(J),J=1,5)                                          TAPE-4
                                                                           TAPE-4
  OUTPUT THE NEW IA-PC FILE
  PRINT 122
  DO 36 I=1,KK
  IPACK = IAPC(I,2)/10000
  IPACK = IPACK*10000 + I
  LIM = IAPC(I,1)
  PRINT 124, IPACK, (IAPC(I,L),L=1,LIM)
  WRITE(62,129) IPACK, (IAPC(I,L),L=3,LIM)
6 CONTINUE                                                                   C0051
```

BEST COPY AVAILABLE

```
 OUTPUT THE PC CONVERSION FILE
  PRINT 110
  DO 40 I=1,1000
  READ(7,103) IPC,IPACK
  IF (EOF,7) 26,28
8 PRINT 112, IPC, IPACK
  WRITE (3,103) IPC,IPACK
0 CONTINUE
6 CONTINUE
  IPC = 99999990    $    IPACK = 9999999
  WRITE (3,103) IPC,IPACK
  OUTPUT THE IA CONVERSION FILE
  PRINT 114
  DO 42 I=1,KK
  IAP = IAPC(I,2)/10000
  IAP = IAP*10000 + I
  PRINT 116, IAPC(I,2), IAP
  WRITE (3,117) IAPC(I,2), IAP
2 CONTINUE
  REWIND 3
  STOP
1 FORMAT (I8,2X,4I10)
3 FORMAT (I8,1X,I7)
7 FORMAT (10X,I7,5X,4(I6,4X))
9 FORMAT (I7,4I6)
0 FORMAT (1H1,35X,*PC CODE CONVERSION TABLE*,///,20X,*8 DIGIT CODE*,
 15X,*SIMULATION CODE*,//)
2 FORMAT (22X,I8,11X,I7)
4 FORMAT (1H1,28X,*IA CODE CONVERSION TABLE*,///,20X,*10 DIGIT CODE*
 1,15X,*SIMULATION CODE*,//)
6 FORMAT (20X,I10,12X,I6)
7 FORMAT (I10,1X,I6)
0 FORMAT (30X,*COMPACTED PC-IA FILE*,///)
2 FORMAT (1H1,30X,*COMPACTED IA-PC FILE*,///)
4 FORMAT (10X,I6,5X,8(I7,3X))
7 FORMAT (I7,3X,4(I6,4X))
9 FORMAT (I6,4X,8(I7,1X))
  END
      SCOPE
0
,10,10000
```

```
NCE,20366,        NOVEMBER
180
,01=(STUDENT-FILE),MT,DA
,7=9
,9=**,MT,DA
,X
PROGRAM ANARCHY
THIS IS THE THIRD PROGRAM OF F D S I M
THIS PROGRAM SIMULATES THE STUDENT FLOW OF THE METER
WRITTEN BY LEININGER AND WILLIAMS, AUGUST-1969
REVISION - OCTOBER-1969

T A P E   A N D   D R U M   R E Q U I R E M E N T S
LU 01 = STUDENT INPUT MASTER-LABELED
LU 15 = GRADUATE FILE ON DRUM
LU 3 = LU 4 = DRUM STUDENT RECORD HOLDERS
LU 6 = BYPASS FOR WEEKLY TAPE OUTPUT
LU 2 = WEEKLY DATA OUTPUT TAPE

LOGICAL UNIT NAMES
   LUNST = LU 3 OR LU 4 - STUDENT RECORDS ON DRUM
   LUIN = INCOMING STUDENT RECORDS ON DRUM -OLD MASTERS-
   LUOUT = OUTGOING STUDENT RECORDS ON DRUM -NEW MASTERS-
LUNDT = LU 6 OR LU 2 - WEEKLY DATA TAPE
      IF LUNDT = 6 NO DATA WILL BE WRITTEN ON THE WEEKLY TAPE
      IF LUNDT = 2 DATA WILL BE WRITTEN ON WEEKLY TAPE

COMMON /C/ CURRNT(800,20), LIMCUR
COMMON /E/ NUMWEEK, NUMAREA, SNUM, NST, STOT                          TEST
COMMON /I/ IADEM(1000,3), IALIM
COMMON /N/ NOGO(15,5)
COMMON /S/ NUMSEM, NUMSTUD(12), SCHST, SCHLOW, SLOP
DIMENSION STREC(500), FINISH(800), GRAD(10)
DIMENSION ITEMP(800,5)
TYPE INTEGER STREC, CURRNT, EPOINT, XPOINT, SNUM, STOT
TYPE INTEGER OVER20, OVERAR, FINISH, GRAD, SLOP, SCHLOW
TYPE INTEGER OPEN

SET UP OVERALL COUNTERS
KWK = 0
SNUM = 0    $    EPOINT = 0    $    STOT = 0
* * * * * L U N D T   A N D   M I D - S T A R T * * * * * * * * * * * * *
LUNDT = 62
R E A D   I N   T H E   S N U M   A N D   M I D
READ 10, SNUM, MID
 FORMAT (2I5)

*  *  IN THE BEGINNING

DUM = RNOG(1)
LUNST = 3
IALIM = 948
DO 95 K=1,1000
5 IADEM(K,1) = K                                                     00054
```

```
  CALL DATAREAD
  IF (MID .EQ. 1) 146,98
6 CONTINUE
  M I D - S T R E A M   S T A R T U P
  DO 140 I=1,800
  BUFFER IN(7,1) (STREC(1),STREC(500))
2 IF(UNIT,7) 132,134,130,991
4 STOT = STOT + 1
  BUFFER OUT(LUNST,1) (STREC(1),STREC(500))
8 IF(UNIT,LUNST) 138,140
0 CONTINUE

0 CONTINUE
  PRINT 822, I, SNUM
  REWIND 7
  E N D   O F   S T A R T U P
  SEMESTER LOOP
8 SNUM = SNUM + 1
  DO 102 I=1,15     $     DO 102 J=1,5
2 NOGO(I,J) = 0
  REWIND 15     $     SET15 = 0.
  EPOINT = XPOINT = SNUM
  TEST FOR RUN COMPLETION
  IF (SNUM .GT. NUMSEM) GO TO 1000
  BRING IN A NEW CLASS
  NST = NUMSTUD(SNUM)
  IF (MID .EQ. 1) GO TO 141
  REWIND 3     $     REWIND 4
  IF (SNUM .EQ. 1) GO TO 112
  DO 112 I=1,STOT
  BUFFER IN(LUNST,1) (STREC(1),STREC(500))
4 IF (UNIT,LUNST) 114,112,922,991
22 PRINT 923
12 CONTINUE
41 CONTINUE
  DO 120 J=1,NST
  K1 = 0
03 BUFFER IN (1,1) (STREC(1),STREC(500))
17 IF (UNIT,1) 117,118,924,901
01 K1 = K1 + 1
  IF (K1 .EQ. 6) 905,907
05 PRINT 933,J
  GO TO 120
07 BACKSPACE 1
  GO TO 903
24 REWIND 01     $     PRINT 925
  GO TO 120
18 STREC(2) = EPOINT
  STREC(5) = HOURS(1)
  STOT = STOT + 1
  IF (STOT .EQ. 801) GO TO 499
  BUFFER OUT (LUNST,1) (STREC(1),STREC(500))
19 IF(UNIT,LUNST) 119,120
20 CONTINUE
```

```
  NEW CLASS APPENDED ONTO LUNST
  HERE ENTER THE SEMESTER ROUTINE
  CONTINUE
  FIRST WEEK OF SEMESTER
  DO 481 K=1,800      $     DO 481 L=1,20
  CURRNT(K,L) = 0
  READ STREC, CREATE CURRNT WITH SEM IA
  REWIND LUNST
  IDROW =0
  BUFFER IN(LUNST,1) (STREC(1),STREC(500))
  IF (UNIT,LUNST) 485,487,520,991
  CONTINUE
  JCOL = 3     $     ITMP = 0

  IDROW = IDROW+1
  CURRNT(IDROW,1) = STREC(1)
  CURRNT(IDROW,2) = STREC(5)
  GO AFTER BOTH SEM R-IA AND E-IA
  N = STREC(6) + 9
  DO 510 J=10,N
  IWKTST = STREC(J)/100
  IWKTST = MOD(IWKTST,100)
  IF (IWKTST .EQ. 14) 502,510
  IRTST = STREC(J)/10000000
  IRTST = MOD(IRTST,10)
  IF (IRTST .EQ. 0 .OR. IRTST .EQ. 5) 503,510
  IF (JCOL .GE. 8) GO TO 511
  JCOL = JCOL+1
  CURRNT(IDROW,JCOL) = STREC(J)
0 CONTINUE
  GO TO 514
  PLANT SEMESTER LENGTH FILL IN COL 20
1 CURRNT(IDROW,20) = 77
  GO TO 501
4 CONTINUE
  GO TO 501
  SUM UP DEMANDS, THEN SORT
  SUM UP
0 LIMCUR = IDROW
  CHECK FOR LIMCUR = STOT
  IF (LIMCUR .NE. STOT) 930,932
0 PRINT 931
2 CONTINUE
  DO 521 J=1,1000     $     DO 521 K=2,3
1 IADEM(J,K) = 0
  IANUM = 0
  DO 530 II=1,IDROW
  DO 529 J=4,20
  IF (CURRNT(II,J) .EQ. 77) GO TO 530
  IF (CURRNT(II,J) .EQ. 0) GO TO 530
  ICUNPK = CURRNT(II,J)/100000000
  IDADD = MOD(ICUNPK,10000)
  IF (IADEM(IDADD,2) .EQ. 0) 525,527
5 IANUM = IANUM + 1
```

```
  527 IADEM(IDADD,2) = IADEM(IDADD,2) + 1
  529 CONTINUE
  530 CONTINUE
      NUMWEEK = 0
C     SCHEDULE IADEM AGAINST RESOURCES
      CALL MATCH1
C     UPDATE THE STRECS AND CURRNTS WITH SCHEDULED IA
      CALL MATCH2
      CALL CURPACK(1)
C
C     ENTER WEEKLY ROUTINE
C
  111 NUMWEEK = NUMWEEK + 1
      IF (NUMWEEK .EQ. 15) GO TO 100
      TIME1 = TIMEF(1)
      REWIND LUNST
      OVERAR = 0.     $     OVER20 = 0.
      IDROW = 0
      IANUM = 0
      DO 621 I=1,1000
      DO 621 J=2,3
  621 IADEM(I,J) =0
C     CLEAR THE DISAPPOINTMENT TALLY
      DO 590 K=1,LIMCUR
  590 CURRNT(K,3) = 0
  600 BUFFER IN(LUNST,1) (STREC(1),STREC(500))
C     READ IN A STREC, COMPUTE OPEN TIME
  602 IF (UNIT,LUNST) 602,601,620,991
  601 IDROW = IDROW + 1
      IF(IDROW .GT. LIMCUR) GO TO 620
      IF (CURRNT(IDROW ,1) .NE. STREC(1)) GO TO 920
C     CHECK FOR 77 IN COL. 20 - NO WEEKLY SCHEDULING DONE
      IF (CURRNT(IDROW,20) .EQ. 77) GO TO 600
C     SUM CURRNT TIMES
      ICHK = 0
      IASUM = 0     $     IFILL = 3
      DO 604 J=4,20
      IF (CURRNT(IDROW ,J) .EQ. 0) GO TO 604
      IFILL = IFILL + 1
      ITIME = CURRNT(IDROW ,J)/10000
      ITIME = MOD(ITIME,100)
      IASUM = IASUM + ITIME
  604 CONTINUE
      OPEN = CURRNT(IDROW,2) - IASUM
C
C     SCHEDULING MECHANISM FOR WEEKLY CYCLES
C
C     CHECK OPEN AGAINST MIN. SCHEDULE TIME (SCHLOW)
      IF (OPEN .LT. SCHLOW) 600,583
  583 AST6 = STREC(6)
      KEY85 = 0
      P = (AST6 * .80) + .5
      IPRC = P
      IF (STREC(7) .GT. IPRC) 581,605
```

```
1 KEY85 = 1   $   GO TO 617
5 INAREA = IVARND(NUMAREA)
  ICHK USED TO PREVENT AN ENDLESS LOOP
  ICHK = ICHK + 1
  IF (ICHK .GT. 10) GO TO 616
  GO TO 617
6 OVERAP = OVERAP + 1      $     GO TO 600
  GET TO THE PROPER AREA+ON THE STREC
7 N = STREC(6) + 9
  DO 612 IST=10,N
  IAREA = STREC(IST)/1000000000000
  IF (KEY85 .EQ. 1) GO TO 607
  IF (IAREA .GT. INAREA) GO TO 622
  IF (IAREA .EQ. INAREA) 607,612
7 ISTATUS = STREC(IST)/10000000
  ISTATUS = MOD(ISTATUS,10)
  IF (ISTATUS .EQ. 7 .OR. ISTATUS .EQ. 1) GO TO 612
  IF (ISTATUS .EQ. 3) GO TO 612
  SEE IF THERE ARE ENOUGH WEEKS
  NWKT = STREC(IST)/100
  NWKT = MOD(NWKT,100)
  NN = 15 - NUMWEEK
  IF (NWKT .GT. NN) 612,609
   GET HOURS PER WEEK
9 JTIME = STREC(IST)/10000
  JTIME = MOD(JTIME,100)

  IF (JTIME .GT. 20) GO TO 613
  JUMP FOR OVER20 HRS/WK

  IOPEN = (OPEN - JTIME) + SLOP
  IF (IOPEN .LT. 0) 612,613
  STORE THE IA IN CURRNT
8 CURRNT(IDROW,IFILL+1) = STREC(IST)
  OPEN =OPEN - JTIME
  PUT A TEMPORARY FLAG IN STATUS
  KPART1 = STREC(IST)/100000000
  KPART2 = MOD(STREC(IST),10000000)
  STREC(IST) = KPART1 * 100000000 + 30000000 + KPART2
  DROP THE CURRNT INTO THE DEMAND LIST
  IDADD = MOD(KPART1,10000)
  IF (IADEM(IDADD,2) .EQ. 0) 625,627
  IANUM = IANUM + 1
  IADEM(IDADD,2) = IADEM(IDADD,2) + 1
  IF (OPEN .LT. SCHLOW) 600,614
  IFILL = IFILL + 1
  IF (IFILL .EQ. 20) 615,612
  CONTINUE
  CONTINUE
  NO IA FOUND IN AREA, GET ANOTHER
  IF (KEY85 .EQ. 1) GO TO 600
  GO TO 605
  OVER20 = OVER20 + 1
  GO TO 600
```

00054

```
END WEEKLY SCHEDULE

CONTINUE
CALL MATCH1
CALL MATCH2
COMPLETION UPDATE, BYPASSING NON-SCHEDULED IA IN CURRNT
DO 709 K=1,800
FINISH(K) = 0
DO 725 IDROW=1,LIMCUR
DO 724 JCOL=4,20
IF (CURRNT(IDROW,JCOL) .EQ.0)  GO TO 724
IF (CURRNT(IDROW,JCOL) .EQ. 77) GO TO 724
SEE IF THE IA IS SCHEDULED OR NOT
ISTAT = CURRNT(IDROW,JCOL)/10000000
ISTAT = MOD(ISTAT,10)
IF (ISTAT .EQ. 5) 712, 714
CURRNT (IDROW,3) = CURRNT(IDROW,3) + 1
GO TO 724
A SCHEDULED EVENT, TEST COMPLETION
IACOUNT = MOD(CURRNT(IDROW,JCOL),100)
IACOUNT = IACOUNT + 1
JWEEK = CURRNT(IDROW,JCOL)/100
JWEEK = MOD(JWEEK,100)
IF (IACOUNT .EQ. JWEEK) 716, 718
NO COMPLETION, UP THE  OUNTER
CURRNT(IDROW,JCOL) = CURRNT(IDROW,JCOL) + 1
GO TO 724
A COMPLETION
IPART1 = CURRNT(IDROW,JCOL)/100000000
IPART2 = MOD(CURRNT(IDROW,JCOL),10000000)
CURRNT(IDROW,JCOL) = IPART1*100000000 + 10000000 + IPART2
FINISH(IDROW) = FINISH(IDROW) + 1
CONTINUE
CONTINUE
UPDATE THE STREC ON DRUM, CHECKING FOR GRADUATION
LUIN = LUNST
REWIND 3    $    REWIND 4
IF (LUNST .EQ. 3) 206,208
LUOUT=4  $  GO TO 209
LUOUT=3
CONTINUE
IDR = 0
KGRAD = 0
IDR = IDR + 1
BUFFER IN (LUIN,1) (STREC(1),STREC(500))
IF (UNIT,LUIN) 721,722,745,991
CONTINUE
IF (STREC(1) .NE. CURRNT(IDR,1)) GO TO 920
STREC(8) = STREC(8) + CURRNT(IDR,3)
DO 730 JCOL=4,20
IF (CURRNT(IDR,JCOL) .EQ. 0) GO TO 730
IF (CURRNT(IDR,JCOL) .EQ. 77) GO TO 730
KCUR = CURRNT(IDR,JCOL)/100000000
```

```
N = STREC(6) + 9
DO 728 JST=10,N
KST = STREC(JST)/100000000
IF (KST .NE. KCUR) 728,726
CONTINUE
PRINT 927, IDR,JCOL,CURRNT(IDR,JCOL)
PRINT 935, (CURRNT(IDR,J),J=1,20)
GO TO 999
STREC(JST) = CURRNT(IDR,JCOL)
CONTINUE
CHECK FOR GRADUATION
STREC(7) = STREC(7) + FINISH(IDR)
IF (NUMWEEK .NE. 14) GO TO 731
ITEMP(IDR,1) = STREC(1)
ITEMP(IDR,2) = STREC(7)                                         ITEMP
ITEMP(IDR,3) = STREC(6)                                         ITEMP
ITEMP(IDR,4) = STREC(8)                                         ITEMP
ITEMP(IDR,5) = STREC(5)                                         ITEMP
CONTINUE                                                        ITEMP
G = STREC(6)
G2 = (G * .95) + .5
IGRAD = G2
IF (STREC(7) .LT. IGRAD) 735, 740
BUFFER OUT (LUOUT,1) (STREC(1),STREC(500))
IF (UNIT,LUOUT) 736,737
CONTINUE
IF (NUMWEEK .EQ. 14) 1010,1020
BUFFER OUT (9,1) (STREC(1),STREC(500))
IF(UNIT,9) 1012,1020
CONTINUE
GO TO 720
CONTINUE
SET UP GRAD FILE RECORD
XPOINT IS THE EXIT POINT
COMPUTE THE NUMBER OF WEEKS IN THE SYSTEM
NWEEKS = (XPOINT-STREC(2))*14 + NUMWEEK
KGRAD = KGRAD + 1
GRAD(1) = STREC(1)
GRAD(2) = STREC(2)
GRAD(3) = XPOINT
GRAD(4) = NWEEKS
GRAD(5) = STREC(3)
GRAD(6) = STREC(5)
GRAD(7) = STREC(6)
GRAD(8) = STREC(8)
WRITE OUT THE FILE ON LU 15
WRITE (15,1115) (GRAD(I),I=1,10)
FORMAT (10I4)
SET15 = 1.
CLEAR OUT CURRNT, BYPASS WRITING OUT THIS STREC
CURRNT(IDR,1) = 0
STOT = STOT - 1
GO TO 720
DONE WITH STREC DRUM UPDATE
```

```
ONTINUE
AVE = 0
L = LIMCUR
O 755 I=1,LIMCUR
AVE = IAVE + FINISH(I)
VE = IAVE
VEFIN = AVE/AL
SUM = 0
DISP = 0
SUM=0   $   NCURKT=0
O 760 I=1,LIMCUR
TSUM = 0
UR = 0
O 758 J=4,20
F (CURRNT(I,J) .EQ. 0) GO TO 758
STAT = CURRNT(IDROW,JCOL)/10000000
STAT = MOD(ISTAT,10)
F (ISTAT .EQ. 5) GO TO 758
T = CURRNT(I,J)/10000
T = MOD(IT,100)
TSUM = ITSUM + IT
UR = KUR + 1
ONTINUE
 = CURRNT(I,2) - ITSUM
UM = ISUM + IS
SUM = JSUM + ITSUM
CURKT = NCURKT + KUR
DISP = IDISP + CURRNT(I,3)
ONTINUE
E = ISUM
EFREE = AVE/AL
ISP = IDISP
ISP = DISP/AL
E = JSUM
ETIME = AVE/AL
E = NCURKT
ECUR = AVE/AL
IP LOGICAL UNITS AROUND, MAKING LUOUT NOW LUIN
NST = LUOUT
CK THE CURRNT VECTOR
LL CURPACK(2)

E WEEKLY CYCLE HAS BEEN COMPLETED
ME2 = TIMEF(1)

EKLY PRINTS
K = KWK + 1
INT 841, NUMWEEK, SNUM
INT 843, KGRAD
INT 871, IANUM
INT 845, OVER20
INT 847, OVERAR
INT 836, AVECUR
INT 849, AVEFIN
```

```
      PRINT 851, AVFREE
      PRINT 838, AVTIME
      PRINT 875, DISP
      PRINT 857
      PRINT 891, NOGO(NUMWEEK+1,1)
      PRINT 892, NOGO(NUMWEEK+1,2)
      PRINT 893, NOGO(NUMWEEK+1,3)
      PRINT 894, NOGO(NUMWEEK+1,4)
      TIME3 = TIME2 - TIME1
      PRINT 820, TIME3
      LUNDT IS LU OF THE WEEKLY DATA TAPE
      WRITE(LUNDT,840) KWK,SNUM,NUMWEEK,KGRAD,IALIM,AVECUR,
     1AVFIN,AVFREE,AVTIME,DISP,(NOGO(NUMWEEK+1,J),J=1,4)


      GO TO 111

      E N D  O F  S E M E S T E R
      CONTINUE
      REWIND 15
      REWIND 9
      SEMESTER PRINT STATEMENTS
      PRINT 861, SNUM
      IF (SET15 .EQ. 1.) 770,771
      CONTINUE
      PRINT 865
      DO 762 I=1,1000
      READ (15,1115) (GRAD(J),J=1,10)
      IF (EOF,15) 763,764
      PRINT 863,I, (GRAD(J),J=1,8)
      CONTINUE
      CONTINUE
      CONTINUE
      IIDR = IDR-1
      PRINT 877
      PRINT 878
      PRINT 879, ((ITEMP(I,J),J=1,5),I=1,IIDR)

      GET THE NEXT SEMESTER ROLLING
      MID = 0
      GO TO 98

      CONTINUE
      REWIND 01
      STOP

      PRINT 921, STREC(1), ((CURRNT(I,J),J=1,3),I=1,LIMCUR)
      GO TO 999
      COMMON DRUM PARITY ERROR FOR BUFFER
      PRINT 992    $    GO TO 999

      WEEKLY FORMATS

      FORMAT (/,10X,*WEEKS CYCLE TIME IN MILLISEC. =*,F10.0)
                                                                  00054
      FORMAT (///,20X,8H* * * * ,*MID-START *,I4,* STUDENTS BUFFERED IN
     1FOR SEMESTER NUMBER *,I3,///)
      FORMAT (10X,*AVERAGE NUMBER OF IA UNDERTAKEN *,F6.2,/)
      FORMAT (10X,*AVERAGE HOURS COMMITTED *,F8.3,/)
      FORMAT (I3,2I2,I3,I4,5F8.2,4I3)
      FORMAT (1H1,20X,*OUTPUT FOR WEEK *,I2,* OF SEMESTER NUMBER *,I2,/)
      FORMAT (//,10X,*NUMBER OF GRADUATES *,I3,/)
      FORMAT (10X,*NUMBER OF CURRNT OVERFLOWS *,I4,/)
      FORMAT (10X,*NUMBER OF EXCESSIVE SEARCHES *,I4,/)
      FORMAT (10X,*AVERAGE NUMBER OF FINISHES *,F8.3,/)
      FORMAT (10X,*AVERAGE FREE TIME-HOURS *,F8.3,/)
      FORMAT (10X,*THIS WEEKS SCHEDULING HANG-UPS*,/)
      FORMAT (1H1,30X,*E N D   O F   S E M E S T E R  *,I4,//)
      FORMAT (10X,I3,5X,8(I4,4X))
      FORMAT (15X,*THIS SEMESTERS GRADUATES*,//)
      FORMAT (10X,*LENGTH OF STUDENT DEMAND ARRAY*,I6,/)
      FORMAT (10X,*AVERAGE NUMBER OF DISAPPOINTMENTS *,F8.3,/)
      FORMAT (//,20X,*S T U D E N T   S T A T U S   D U M P*,/)
      FORMAT (6X,*STUDENT*,2X,*COMPLETIONS*,2X,*TOTAL*,2X,*DISAPPOINTMEN
     1TS*,2X,*HOUR AVAIL.*,/,7X,*I.D.*,5X,*TO DATE*,15X,*TO DATE*,//)
      FORMAT (7X,I4,7X,I4,6X,I4,8X,I4,12X,I4)
      FORMAT (15X,*SEM-LENGTH EXCESS    *,4X,I6,/)
      FORMAT (15X,*INADEQUATE DEMAND    *,4X,I6,/)
      FORMAT (15X,*INADEQUATE TEACHERS *,4X,I6,/)
      FORMAT (15X,*NO CLASSROOM SPACE  *,4X,I6,/)

      E R R O R   F O R M A T S
      FORMAT (//,5X,8H* * * * ,*F A T E L   E R R O R - CURRNT/STREC OUT
     1OF SEQUENCE*,/,10X,*STREC IS*,I4,/,(10X,3I10))
      FORMAT (//,5X,8H* * * * , *EOF HIT ON LUNST RUNDOWN FOR NEW STUDEN
     1TS INCLUSION*,//)
      FORMAT (/,5X,8H* * * * ,*STUDENT TAPE REWOUND*,/)
      FORMAT (//,5X,8H* * * * ,*THIS CURRNT COULD NOT BE MATCHED TO A ST
     1REC    *,3(I14,2X),/)
      FORMAT (//,5X,8H* * * * , *SEMESTER SETUP OF CURRNT HAS LIMCUR .NE
     1. TO TOTAL STUDENTS ON DRUM*,//)
      FORMAT (/,10X,8H* * * * ,*STUDENT *,I4,* BYPASSED DUE TO PARITY*)
    5 FORMAT (//,10X,5(I14,2X))
      FORMAT (//,5X,8H* * * * ,*F A T E L   E R R O R - DRUM PARITY*,//)
      REWIND 01
      STOP
      END                                                               00042
```

```
  SUBROUTINE DATAREAD
  COMMON /A/ NUMRES, ICRES(15,14), ICRESCAP(15), NIA,
 1IAREQ(1000,10), TMAX
  COMMON /E/ NUMWEEK, NUMAREA, SNUM, NST, STOT
  COMMON /B/ NUMINST, NTEACH(4,14,14)
  COMMON /S/ NUMSEM, NUMSTUD(12), SCHST, SCHLOW, SLOP
  COMMON /T/ STIMDST(4)
  COMMON/W/IWDRES(15,2),IWDAREA(14,2),IWDINS(4,2),IWDIS(5,2)
  READS DATA FOR EDSIM2 PROGRAM
  INTEGER SCHLOW, SLOP
  DIMENSION IREQHLD(10)
  INPUT NUMBER OF SEMESTERS MODEL TO RUN, MAXIMUM=12
  READ(60,101) NUMSEM
  INPUT NUMBER OF STUDENTS ENTERING SYSTEM EACH SEMESTER,MAXIMUM=300
  READ(60,101) (NUMSTUD(I),I=1,NUMSEM)
  INPUT SCHEDULE VARIABLE,MUST=1 OR GREATER
  READ(60,102) SCHST
  SCHLO=LOWER LIMIT OF SCHEDULE PROGRAM
  READ(60,101)SCHLOW
  SLOP=ALLOWED VARIANCE IN IA SCHEDULING
  READ(60,101) SLOP
  STUDENT TIME DIST   (1)=10 (2)=15  (0)=20  (J)=25  CUMULATIVE DST
  READ(60,102)(STIMDST(I),I=1,4)
  INPUT TMAX,MAXIMUM TEACHING FOR SEMESTER LENGTH IA, LESS THAN 1
  READ (60,102) TMAX
  INPUT TOTAL NUMBER OF IA, MAXIMUM = 1700
  READ (60,101) NIA
  INPUT RESOURCE MATRIX, NUMBER RESOURCE TYPES MAXIMUM = 4
  READ (60,101) NUMRES
  INPUT RESOURCE CAPACITY
  READ (60,101)(ICRESCAP(I),I=1, NUMRES)
  INPUT IAREQ MATRIX (DIMENSION 1700 X 10)
  DO 3 I=1,NIA  $  READ 103,(IREQHLD(J),J=1,10)
  IADD=MOD(IREQHLD(1),10000)
  IF(IREQHLD(2).EQ.0)IREQHLD(2)=1
  IF(IREQHLD(3).GT.14)IREQHLD(3)=14
  IF(IREQHLD(3).LT.1)IREQHLD(3)=1
  IF(IREQHLD(2).GT.50)IREQHLD(2)=50
  IF (IREQHLD(4) .EQ. 0) IREQHLD(4)=5
  IF (IREQHLD(5) .EQ. 0) IREQHLD(5)=5
  IF(IREQHLD(6).EQ.0)IREQHLD(5)=999
  IF(IREQHLD(7).GT.25)IREQHLD(7)=25
  IF(IREQHLD(9).GT.25)IREQHLD(9)=25
  IF(IREQHLD(10).GT.15)IREQHLD(10)=15
  IF(IREQHLD(10).LE.0)IREQHLD(10)=1
  IF(IREQHLD(6).GT.3)IREQHLD(6)=3
  IF(IREQHLD(8).GT.3)IREQHLD(8)=3                        FU+ R)  -+LD
  IF(IREQHLD(8).LT.0)IREQHLD(8)=1
  IF(IREQHLD(5).EQ.1)1,2
1 IREQHLD(5)=5  $  IREQHLD(7)=5
2 CONTINUE
  IF(IREQHLD(8).EQ.0)GO TO 5
  IF(IREQHLD(6).NE.IREQHLD(8))GO TO 5                              000
```

```
IREQHLD(7)=IREQHLD(7)+IREQHLD(9)
IF(IREQHLD(7).GT.25)IREQHLD(7)=25
IREQHLD(8)=0   $   IREQHLD(9)=0
IF(IREQHLD(5).LT.999)GO TO 6
IF(IREQHLD(7).EQ.0)GO TO 6
IF(IREQHLD(5).GT.800)IREQHLD(5)=800
IF(IREQHLD(6).EQ.0)GO TO 7
IK=IREQHLD(10)
IF(IREQHLD(5) .GT. ICRESCAP(IK)) IREQHLD(5)=ICRESCAP(IK)
DO 8 K=1,10
IAREQ(IADD,K)=IREQHLD(K)
CONTINUE
INPUT NUMBER OF WEEKS PER SEMESTER, MAXIMUM = 1+
READ (60,101) NOWEEK
INPUT RESOURCE MATRIX
DO 10 I=1, NUMRES
READ (60,101)(ICRES(I,J),J=1, NOWEEK)
INPUT TEACHING RESOURCE MATRIX, NUMINST, MAXIMUM = 4
READ (60,101) NUMINST
INPUT NUMBER OF AREAS, MAXIMUM =14
READ (60,101) NUMAREA
INPUT TEACHING MATRIX
DO 15 I= 1, NUMINST
DO 15 K = 1, NUMAREA
READ(60,101)(NTEACH(I,J,K),J=1,NOWEEK)
INPUT OUTPUT HEADINGS
DO 20 I=1,NUMRES
READ 104,(IWDRES(I,J),J=1,2)
DO 25 I=1,NUMAREA
READ 104,(IWDAREA(I,J),J=1,2)
DO 27 I=1,NUMINST
READ 104,(IWDINS(I,J),J=1,2)
DO 28I=1,5
READ 104,(IWDIS(I,J),J=1,2)
PRINT 115   $   PRINT 120   $   PRINT 125   $   PRINT 130
DO 30 I=1,NUMRES
PRINT 150,(IWDRES(I,J),J=1,2)
PRINT 110,(ICRES(I,J),J=1,NOWEEK)
PRINT 135   $   PRINT 140    $   PRINT 145
DO 35 I=1,NUMINST
PRINT 150,(IWDINS(I,M),M=1,2)
DO 35 K=1,NUMAREA
PRINT 150,(IWDAREA(K,N),N=1,2)
PRINT 110,(NTEACH(I,J,K),J=1,NOWEEK)
DATAREAD FORMAT STATEMENTS
FORMAT(15I4)
FORMAT (15F4.0)
FORMAT(I6,8X,2I2,6X,I2,I3,1X,I1,I2,I1,I2,4X,I2)
FORMAT(2A8)
FORMAT(14(2X,I5))
FORMAT(37X,*RESOURCE LOADINGS FOR RUN*/)
FORMAT(39X,*AVAILABLE CLASSROOMS*/)
FORMAT(44X,*TYPE X WEEKS*/)
FORMAT(31X,*CAPACITY X HOURS AVAILABLE PER WEEK*//)
```

```
5 FORMAT(//39X,*AVAILABLE INSTRUCTION*//)
0 FORMAT(40X,*AREA X WEEKS X TYPE*/)
5 FORMAT(32X,*NO. FACULTY MEMBERS X TEACHING LOAD*//)
0 FORMAT(/41X,2A8,/)
  RETURN     $    END
                                                                  000
```

BEST COPY AVAILABLE

```
SUBROUTINE MATCH1
MATCHES IA DEMANDED WITH AVAILABLE RESOURCES
NTEGER SNUM, STOT
COMMON /A/ NUMRES, ICRES(15,14), ICRESCAP(15), NIA,
IAREQ(1000,10), TMAX
COMMON /B/ NUMINST, NTEACH(4,14,14)
COMMON /E/ NUMWEEK, NUMAREA, SNUM, NST, STOT
COMMON /I/ IADEM(1000,3), IALIM                                           TEST
COMMON/W/IWDRES(15,2),IWDAREA(14,2),IWDINS(4,2),IWDIS(5,2)
DIMENSION NTMAX(4,14),NTSUM(4,14),NTEWORK(4,14,14),IRESWK(15,14)
DIMENSION NUMDIS(5,14,11),NEXD(14)
DIMENSION ICRSTAT(15,14),IRMEAN(15),STD(15),IRLOW(15),IRHIGH(15)
DIMENSION ISSQ(15),AVG(15),ITESTAT(4,14,14)
CHECK TO DETERMINE IF BEGINNING NEW SEMESTER
NOWEEK = 14
 RELOAD TEACHING RESOURCE MATRIX
IF (NUMWEEK .GT. 0) 20,2
DO 5 I=1, NUMINST
DO 5 J=1, NOWEEK
DO 5 K=1, NUMAREA
NTEWORK(I,J,K)=NTEACH(I,J,K)
RELOAD TMAX MATRIX
DO 7 I=1, NUMINST
DO 7 J=1, NUMAREA
ATEACH = NTEACH (I,1,J)
NTMAX(I,J) = ATEACH * TMAX
NTSUM(I,J) = 0
RELOAD CLASSROOM RESOURCE MATRIX
DO 9 I = 1, NUMRES
DO 9 J = 1, NOWEEK
IRESWK (I,J) = ICRES (I,J)
SUM TEACHING REQUIREMENT
DO 11 I=1,5  $  DO 11 J=1,14  $  DO 11 K=1,11
NUMDIS(I,J,K)=0
DO 12 I=1,14
NEXD(I)=0
DO 15 I = 1, IALIM
IF (IADEM(I,2) .EQ. 0) GO TO 15
IE = IADEM(I,1)
ID = IAREQ(IE,1)/10000
IF (IAREQ(IE,5).EQ.999) GO TO 15
DO 15 K = 6,8,2
IF (IAREQ(IE,K).EQ.00) GO TO 15                              I 3 8+ I-)-8
NTSUM(IG,ID) = NTSUM(IG,ID) + IAREQ(IE,K+1)
IF (NTSUM(IG,ID).LE.NTMAX(IG,ID)) GO TO 15
NTSUM(IG,ID)=NTSUM(IG,ID)-IAREQ(IE,K+1)
IADEM(I,2)=0  $  IADEM(I,3)=1
 CODE  1  =  TMAX EXCEEDED
IADEM(I,2)=0  $  IADEM(I,3)=1
CONTINUE
TMAX AND SEMESTER SETUP ROUTINES COMPLETED
ISDM = 0
DO 25 I = 1, IALIM
```

```
5 ISDM = ISDM + IADEM (I,2)
  NXWEEK = NUMWEEK
  IF (NXWEEK.EQ.0) NXWEEK = 1
  DO 80 IA = 1, IALIM
  IF (IADEM (IA,3).EQ.1) GO TO 80
  IF (IADEM(IA,2) .EQ. 0) GO TO 80
  IE = IADEM(IA,1)
  ID = IAREQ(IE,1)/10000
  TEST TO SEE IF IA IS SELF STUDY
  IF (IAREQ(IE,5).EQ.999) GO TO 62
   TEST TO SEE IF STUDENT DEMAND ADEQUATE
  IF(IADEM(IA,2).LT.IAREQ(IE,4)) 35 , 38
   DEMAND INADEQUATE CODE = 2
5 NUMDIS(2,NXWEEK,ID)=NUMDIS(2,NXWEEK,ID)+IADEM(IA,2)
  IADEM(IA,2)=0  $  IADEM(IA,3)=2  $  GO TO 80
   TEST IF DEMAND EXCEEDS CAPACITY
8 IF(IADEM(IA,2).GT.IAREQ(IE,5))40,42
0 NOVER=IADEM(IA,2)-IAREQ(IE,5)
  IADEM(IA,2)=IAREQ(IE,5)
  NUMDIS(5,NXWEEK,ID)=NUMDIS(5,NXWEEK,ID)+NOVER
  NEXD(NXWEEK)=NEXD(NXWEEK)+1
  BEGIN TO CHECK RESOURCES
   CHECK TO SEE +F TEACHE- AVA+LABLE
2 II = IAREQ(IE,6)
  IF (IAREQ(IE,7).GT.NTEWORK(II,NXWEEK,ID)) 44,48
  LACK OF TEACHERS, CODE = 3
4 NUMDIS(3,NXWEEK,ID)=NUMDIS(3,NXWEEK,ID)+IADEM(IA,2)
  IADEM(IA,2)=0  $  IADEM(IA,3)=3  $  GO TO 80
8 IF(IAREQ(IE,8).EQ.0) GO TO 54
  II = IAREQ(IE,8)
  IF(IAREQ(IE,9).GT.NTEWORK(II,NXWEEK,ID)) 50,54
  LACK OF TEACHERS   CODE = 3
0 NUMDIS(3,NXWEEK,ID)=NUMDIS(3,NXWEEK,ID)+IADEM(IA,2)
  IADEM(IA,2)=0  $  IADEM(IA,3)=3  $  GO TO 80
  CHECK TO SEE IF CLASSROOM AVAILABLE
4 IX = IAREQ(IE,10)
  NET=ICRESCAP(IX)*IAREQ(IE,2)
  IF (NET.GT. IRESWK(IX,NXWEEK))   60,  68
  LACK OF CLASSROOM CODE  =  4
0 NUMDIS(4,NXWEEK,ID)=NUMDIS(4,NXWEEK,ID)+IADEM(IA,2)
  IADEM(IA,2)=0  $  IADEM(IA,3)=4  $  GO TO 80
  CHECK CLASSROOM FOR SELF STUDY
2 IX = IAREQ(IE,10)
  IH = IADEM(IA,2) * IAREQ(IE,2)
  IF(IH.GT.IRESWK(IX,NXWEEK)) 64, 68
4 NZ=IADEM(IA,2)-1
  DO 66 IZ = 1, NZ
  IH = (IADEM(IA,2)-IZ)*IAREQ(IE,2)
  IF(IH.LT.IRESWK(IX,NXWEEK)) GO TO 67
6 CONTINUE
  NUMDIS(4,NXWEEK,ID)=NUMDIS(4,NXWEEK,ID)+IADEM(IA,2)
  IADEM(IA,2) = 0   $   IADEM(IA,3) = 4   $   GO TO 80
7 IADEM(IA,2) = IADEM(IA,2) -IZ
  NUMDIS(4,NXWEEK,ID)=NUMDIS(4,NXWEEK,ID)+IZ
```

```
 ISDM = ISDM + IADEM (I.2)
 NXWEEK = NUMWEEK
 IF (NXWEEK.EQ.0) NXWEEK = 1
 DO 80 IA = 1, IALIM
 IF (IADEM (IA.3).EQ.1) GO TO 80
 IF (IADEM(IA.2) .EQ. 0) GO TO 80
 IE = IADEM(IA,1)
 ID = IAREQ(IE,1)/10000
 TEST TO SEE IF IA IS SELF STUDY
 IF (IAREQ(IE.5).EQ.999) GO TO 62
  TEST TO SEE IF STUDENT DEMAND ADEQUATE
 IF(IADEM(IA.2).LT.IAREQ(IE.4)) 35 , 38
  DEMAND INADEQUATE CODE = 2
 NUMDIS(2.NXWEEK,ID)=NUMDIS(2.NXWEEK,ID)+IADEM(IA.2)
 IADEM(IA.2)=0  $  IADEM(IA.3)=2  $  GO TO 80
  TEST IF DEMAND EXCEEDS CAPACITY
 IF(IADEM(IA.2).GT.IAREQ(IE.5))40.42
 NOVER=IADEM(IA.2)-IAREQ(IE.5)
 IADEM(IA.2)=IAREQ(IE.5)
 NUMDIS(5.NXWEEK,ID)=NUMDIS(5.NXWEEK,ID)+NOVER
 NEXD(NXWEEK)=NEXD(NXWEEK)+1
 BEGIN TO CHECK RESOURCES
  CHECK TO SEE +F TEACHE- AVA+LABLE
 II = IAREQ(IE.6)
 IF (IAREQ(IE.7).GT.NTEWORK(II.NXWEEK,ID)) 44.48
 LACK OF TEACHERS, CODE = 3
 NUMDIS(3.NXWEEK,ID)=NUMDIS(3.NXWEEK,ID)+IADEM(IA.2)
 IADEM(IA.2)=0  $  IADEM(IA.3)=3  $  GO TO 80
 IF(IAREQ(IE.8).EQ.0) GO TO 54
 II = IAREQ(IE.8)
 IF(IAREQ(IE.9).GT.NTEWORK(II.NXWEEK,ID)) 50.54
 LACK OF TEACHERS   CODE = 3
 NUMDIS(3.NXWEEK,ID)=NUMDIS(3.NXWEEK,ID)+IADEM(IA.2)
 IADEM(IA,2)=0  $  IADEM(IA.3)=3  $  GO TO 80
 CHECK TO SEE IF CLASSROOM AVAILABLE
 IX = IAREQ(IE.10)
 NET=ICRESCAP(IX)*IAREQ(IE.2)
 IF (NET.GT. IRESWK(IX.NXWEEK))   60,  68
 LACK OF CLASSROOM CODE  =  4
 NUMDIS(4.NXWEEK,ID)=NUMDIS(4.NXWEEK,ID)+IADEM(IA.2)
 IADEM(IA.2)=0  $  IADEM(IA.3)=4  $  GO TO 80
 CHECK CLASSROOM FOR SELF STUDY
 IX = IAREQ(IE.10)
 IH = IADEM(IA.2) * IAREQ(IE.2)
 IF(IH.GT.IRESWK(IX.NXWEEK)) 64, 68
 NZ=IADEM(IA.2)-1
 DO 66 IZ = 1, NZ
 IH = (IADEM(IA.2)-IZ)*IAREQ(IE.2)
 IF(IH.LT.IRESWK(IX.NXWEEK)) GO TO 67
 CONTINUE
 NUMDIS(4.NXWEEK,ID)=NUMDIS(4.NXWEEK,ID)+IADEM(IA.2)
 IADEM(IA.2) = 0   $   IADEM(IA.3) = 4   $   GO TO 80
 IADEM(IA.2) = IADEM(IA.2) -IZ
 NUMDIS(4.NXWEEK,ID)=NUMDIS(4.NXWEEK,ID)+IZ
```

```
C        ALLOCATE RESOURCES IA SCHEDULED- TEACHERS
      68 IY = IAREQ (IE,3)
         KONT = NXWEEK + IY - 1
         IF (IAREQ(IE,5) .EQ. 999) GO TO 76
         II=IAREQ(IE,6)
         DO 69 IJ = NXWEEK, KONT,1
      69 NTEWORK(II,IJ,ID)=NTEWORK(II,IJ,ID)-IAREQ(IE,7)
         IF (IAREQ(IE,8).EQ.0) GO TO 72
         II=IAREQ(IE,8)
         DO 70 IJ = NXWEEK, KONT, 1
      70 NTEWORK(II,IJ,ID)=NTEWORK(II,IJ,ID)-IAREQ(IE,9)
C        ALLOCATE CLASSROOMS TO TOUGHT IA
      72 DO 73 IJ = NXWEEK,KONT,1
      73 IRESWK(IX,IJ) = IRESWK(IX,IJ) - NET
         GO TO 80
C        ALLOCATE CLASSROOM TO SELF STUDY IA
      76 DO 78 IJ = NXWEEK, KONT, 1
      78 IRESWK(IX,IJ) = IRESWK(IX,IJ) - IH
      80 CONTINUE
C        SUM SATISFIED STUDENTS
         NSTSAT = 0
         DO 82 I = 1 , IALIM, 1
      82 NSTSAT = NSTSAT + IADEM(I,2)
C        OUTPUT STATISTICS FOR SCHEDULING PERIOD
         IF(NUMWEEK.EQ.0)GO TO 100
         IF(NUMWEEK.EQ.14)GO TO 200   $   RETURN
     100 PRINT 121, SNUM
         PRINT 122, NST
         NOLD = STOT-NST
         PRINT 123, NOLD
         RETURN
     200 DO 205 I=1,15
         IRMEAN(I)=0  $  STD(I)=0  $  ISSQ(I)=0   $  AVG(I)=0.0
         IRLOW(I)=ICRES(I,1)-IRESWK(I,1)
         IRHIGH(I)=ICRES(I,14)-IRESWK(I,14)
         DO 205 J=1,14
     205 ICRSTAT(I,J)=ICRES(I,J)-IRESWK(I,J)
         DO 210 I=1,15  $  DO 208 J=1,14
         IF(ICRSTAT(I,J).LT.IRLOW(I))IRLOW(I)=ICRSTAT(I,J)
         IF(ICRSTAT(I,J).GT.IRHIGH(I))IRHIGH(I)=ICRSTAT(I,J)
     208 IRMEAN(I)=IRMEAN(I)+ICRSTAT(I,J)
         IRMEAN(I)=IRMEAN(I)/14
         A=IRMEAN(I)  $  B=ICRES(I,J)
     210 AVG(I)=A/B
         DO 220 I=1,15  $  DO 218 J=1,14
     218 ISSQ(I)=ISSQ(I)+((IRMEAN(I)-ICRSTAT(I,J))**2)
         AVAR=ISSQ(I)/14
     220 STD(I)=SQRT(AVAR)
         PRINT 126  $  PRINT 230  $  PRINT 231
     230 FORMAT(25X,*RESOURCES SCHEDULED DURING SEMESTER-FIRST LINE*,/)
     231 FORMAT(22X,*RESOURCES NOT SCHEDULED DURING SEMESTER-SECOND LINE*/)
         DO 240 I=1,15
         PRINT 127,(IWDRES(I,M),M=1,2) $ PRINT 128,(ICRSTAT(I,J),J=1,14)
     240 PRINT 128,(IRESWK(I,J),J=1,14)
```

```
      PRINT 241,SNUM   $  PRINT 242
  241 FORMAT(///,2X,*RESOURCE STATISTICS FOR SEMESTER NUMBER*,1X,I2,/)
  242 FORMAT (5X,*RESOURCE*,10X,*CAPACITY*,5X,*HIGH     LOW      MEAN       *
     1,*STD DEV*,5X,*AVERAGE*,/)
      DO 245 I=1,15
      PRINT 246,(IWDRES(I,J),J=1,2),ICRESCAP(I),IRHIGH(I),IRLOW(I),IRMEA
     1 N(I),STD(I),AVG(I)
  245 CONTINUE
  246 FORMAT(2X,2A8,11X,I4,3X,I4,3X,I4,2X,I6,3X,F8.2,3X,F8.2)
      DO 400 J=1,3
      DO 305 I=1,11
      IRMEAN(I) = 0
      STD(I)=0.0  $  ISSQ(I)=0
      IRLOW(I) = NTEACH(J,1,I) - NTEWORK(J,1,I)
      IRHIGH(I) = NTEACH(J,14,I) - NTEWORK(J,14,I)
      DO 305 K=1,14
  305 ITESTAT(J,K,I) = NTEACH(J,K,I) - NTEWORK(J,K,I)
      DO 310 I=1,11
      DO 308 K=1,14
      IF (IRLOW(I) .GT. ITESTAT(J,K,I)) IRLOW(I)=ITESTAT(J,K,I)
      IF (IRHIGH(I) .LT. ITESTAT(J,K,I)) IRHIGH(I)= ITESTAT(J,K,I)
  308 IRMEAN(I) = IRMEAN(I) + ITESTAT(J,K,I)
      A = IRMEAN(I)     $     B = NTEACH(J,1,I)
      A = A/14.
      IRMEAN(I) = IRMEAN(I)/14
  310 AVG(I)=A/B
      DO 320 I=1,11     $     DO 318 K=1,14
  318 ISSQ(I)=ISSQ(I)+((IRMEAN(I)-ITESTAT(J,K,I))**2)
      AVAR = ISSQ(I)
      AVAR = AVAR/14.
  320 STD(I) = SQRT(AVAR)
      PRINT 125     $     PRINT 230     $     PRINT 231
      PRINT 127, (IWDINS(J,M),M=1,2)
      DO 330 I=1,11
      PRINT 127, (IWDAREA(I,N),N=1,2)
      PRINT 128, (ITESTAT(J,L,I),L=1,14)
  330 PRINT 128, (NTEWORK(J,L,I),L=1,14)
      PRINT 341, SNUM     $     PRINT 342
      DO 345 I=1,11
  345 PRINT 346, (IWDAREA(I,N),N=1,2),IRHIGH(I),IRLOW(I),IRMEAN(I),
     1STD(I) , AVG(I)
  400 CONTINUE
  341 FORMAT (///,2X,*INSTRUCTION STATISTICS FOR SEMESTER NUMBER *,I2,/)
  342 FORMAT (5X,*RESOURCE*,10X,*HIGH       LOW       MEAN       STD DEV
     1 AVERAGE*,/)
  346 FORMAT (2X,2A8,5X,I4,2X,I4,3X,I4,3X,F8.2,3X,F8.2)
  125 FORMAT (///,40X,*T E A C H E R  M A T R I X*,/)
      PRINT 133
      DO 116 I=1,11
      PRINT 127,(IWDAREA(I,N),N=1,2)
      DO 116 J=1,5
   16 PRINT 132,(IWDIS(J,K),K=1,2),(NUMDIS(J,L,I),L=1,14)
   21 FORMAT (1H1,35X,*STARTING SEMESTER NUMBER*,1X, I2,/)
   22 FORMAT (35X,I3,* STUDENTS ENTERING SYSTEM*,/)
```

```
123 FORMAT (33X,I3,* STUDENTS REMAINING IN SYSTEM*,/)
126 FORMAT (/,42X,*RESOURCE MATRIX*,/)
127 FORMAT(41X,2A8,/)
128 FORMAT (14(2X,I5))
130 FORMAT (1H1,27X,*RESOURCE SUMMARY, END OF SEMESTER NUMBER *,I2,/)
131 FORMAT (27X,*RESULTS OF SCHEDULING FOR THE ENTIRE SEMESTER*,/)
132 FORMAT(3X,2A8,2X,14(2X,I5)/)
133 FORMAT(1H1,44X,*DISAPPOINTMENT FILE*,//)
    RETURN
    END
```

```
      SUBROUTINE MATCH2
C     UPDATES CURRNT FROM IADEM RESULTS
C     REWRITTEN 11/16 TO MAKE IADEM ARRAY AN ADDRESS LOOK-UP
C
      COMMON /C/ CURRNT(800,20), LIMCUR
      COMMON /E/ NUMWEEK, NUMAREA, SNUM, NST, STOT
      COMMON /I/ IADEM(1000,3), IALIM
      COMMON /N/ NOGO(15,5)
      INTEGER CURRNT
      DO 50 IROW = 1,LIMCUR
      DO 48 JCOL=4,20
      IF (CURRNT(IROW,JCOL) .EQ. 0) GO TO 48
      ISTAT = CURRNT(IROW,JCOL)/10000000
      ISTAT = MOD(ISTAT,10)
      IF (ISTAT .EQ. 7) GO TO 48
      IACODE = CURRNT(IROW,JCOL)/100000000
      IDADD = MOD(IACODE,10000)
      IF (IADEM(IDADD,2) .GT. 0) 40,45
   40 IPART = MOD(CURRNT(IROW,JCOL),10000000)
      IREPLAC = IACODE*100000000 + 70000000 + IPART
      CURRNT(IROW,JCOL) = IREPLAC
C     DROP ONE STUDENT FROM SCH. LIST - TO GET EXCESS TAKEN CARE OF
      IADEM(IDADD,2) = IADEM(IDADD,2) - 1
      GO TO 48
   45 IPART = MOD(CURRNT(IROW,JCOL),10000000)
      IREPLAC = IACODE*100000000 + 50000000 + IPART
  508 CURRNT(IROW,JCOL) = IREPLAC
   48 CONTINUE
   50 CONTINUE
      NWK = NUMWEEK + 1
      DO 70 K=1,IALIM
      KTYPE = IADEM(K,3)
      IF (KTYPE .EQ. 0) GO TO 70
      NOGO(NWK,KTYPE) = NOGO(NWK,KTYPE) + 1
   70 CONTINUE
      RETURN    $    END
   FUNCTION HOURS(X)
   GIVES BACK STUDENT HOURS PER WEEK
   CURRENTLY 4 POSSIBLE HRS/WK  -  8/4
   COMMON /T/ STIMDST(4)
   DIMENSION HR(4)
   SET THE HOUR COMBINATIONS
   HR(1)=20.  $  HR(2)=25.  $  HR(3)=30.  $  HR(4)=35.
   RAND = RNOG(2)
   RAND = RAND/100.
   DO 10 J=1,4
   IF (RAND .LE. STIMDST(J)) 12,10
10 CONTINUE
   GO TO 15
12 HOURS = HR(J)
   RETURN
15 HOURS = 15.
   PRINT 20
20 FORMAT (/,5X,*ERROR IN HOURS --- DIST. FELL THROUGH*,/)
   RETURN    $    END
                                                                  0001
```

```
      SUBROUTINE CURPACK(I)
      ROUTINE TO PACK THE CURRNT VECTOR, DROPPING ANY GRADS COMPLETELY
      AND REMOVING ANY COMPLETED OR UNSCHEDULED IA
      COMMON /C/ CURRNT(800,20), LIMCUR
      COMMON /E/ NUMWEEK, NUMAREA, SNUM, NST, STOT
      DIMENSION KSAV(20)
      INTEGER CURRNT
      DO 50 L=1,LIMCUR
      KK = 0    $     K77SET = 0
      DO 18 M=1,20
   18 KSAV(M) = 0
      IF (CURRNT(L,20) .EQ. 77) K77SET=1
   30 DO 40 K=4,19
      IF (CURRNT(L,K) .EQ. 0) GO TO 40
      IDMD = CURRNT(L,K)/10000000
      IDMD = MOD(IDMD,10)
      IF (IDMD .EQ. 7) 35, 33
   33 CURRNT(L,K) = 0     $     GO TO 40
   35 KK = KK + 1
      KSAV(KK) = CURRNT(L,K)
   40 CONTINUE
      KLIM = KK +4
      DO 45 J=4,KLIM
   45 CURRNT(L,J) = KSAV(J-3)
      KLIM = KLIM+1
      DO 47 J=KLIM,20
   47 CURRNT(L,J) = 0
      IF (K77SET .EQ. 1) CURRNT(L,20)=77
   50 CONTINUE
      IF (I .EQ. 1) 52,54
   54 CONTINUE
      SECOND PHASE - DROP 0 SLOTS DUE TO GRADUATION
      LIM1 = LIMCUR
      IR = 0
  250 IR = IR + 1
      IF (IR .GT. LIM1) GO TO 300
      IF (CURRNT(IR,1) .EQ. 0) 260,250
  260 LIM2 = LIM1 - 1
      DO 270 IRR=IR,LIM2
      DO 265 J=1,20
  265 CURRNT(IRR,J) = CURRNT(IRR+1,J)
  270 CONTINUE
      IR = IR - 1
      LIM1 = LIM1 - 1
      GO TO 250
  300 CONTINUE
      ZERO PACK TAIL END OF CURRNT - RESET LIMCUR
      LIM3 = LIM1 + 1
      DO 305 L=LIM3,LIMCUR
      DO 305 J=1,20
  305 CURRNT(L,J) = 0
      LIMCUR = LIM1
   52 RETURN     $     END
```

COO

```
      FUNCTION IVARND(NSLOTS)
      FUNCTION TO GIVE A RANDOM NUMBER BETWEEN 1 AND THE INPUTED
      VALUE OS NSLOTS
      IF (NSLOTS .LE. 1) 8,10
    8 IVARND = 1
      RETURN
   10 SLOT = NSLOTS
      DIV = 1./SLOT
      RAND = RNOG(7)
      HOLD = RAND/DIV
      IVA = HOLD+1.
      IF (IVA .GT. NSLOTS) 15,20
   15 PRINT 30, NSLOTS, IVA
      IVA = NSLOTS
   20 IVARND = IVA
   30 FORMAT (//,5X,*ERROR IN IVARND - VALUE RETURNED IS NOT PROPER*,
     15X,*NSLOTS =*,I5,5X,*IVA =*,I5)
      RETURN    $    END
                                                                     0001
      FUNCTION RNOG(ICODE)
C     REVISED 5/11 WITH EXPANDED CODE
C        ICODE = 1             SET SEED
C        ICODE = 2 THRU 5      RETURN INTEGER, LENGTH OF ICODE
C        ICODE = 7             RETURN REAL 0 TO 1, LENGTH 5 DIGITS
      IF (ICODE .GT. 1) GO TO 15
      IA = 998877
   15 IFAC = IA * 455470314
      IB = MOD(IFAC,214783646)
      IC = IABS(IB)/100
C     NOW SPLIT VIA CODE
      IF (ICODE .EQ. 1) 17,19
   17 INDEX = 100     $     GO TO 25
   19 IF (ICODE .EQ. 7) GO TO 30
      INDEX = 10**ICODE
   25 RNOG = MOD(IC,INDEX)
      GO TO 35
   30 R = MOD(IC,100000)
      RNOG = R/100000
   35 IA = IB
      RETURN  $  END
```

```
TIME,10
EQUIP,3=(CODE-CONVERT),MT,DA
FTN,L,X
      PROGRAM CVTREQ
C     CONVERTS THE IAREQ FILE
C     WRITTEN BY LEININGER AND WILLIAMS  -  AUGUST 69
C
C     NEEDS THE IA-PC CONVERSION CODE TAPE
C     LU 3 = CONVERSION TAPE
C     LU 5 = TEMP DRUM FILE OF ERROR RECORDS
C
      DIMENSION IACONVT(1500,3), IAREQ1(10), IAREQ2(10)
      INTEGER OLD
      KOUNT = 0
      DO 8 I=1,1500
    8 IACONVT(I,3) = 0
      PRINT 120
C     READ THE CONVERSION TAPE TO GET TO THE IA CODES
      DO 12 I=1,1000
      READ (3,103) IPC,IA
      IF (IPC .NE. 99999999) 12,14
   14 IF (IA .EQ. 9999999) 16,91
   12 CONTINUE
   16 DO 20 IFILE=1,2000
      READ (3,104) OLD, NEW
      IF (EOF,3) 21,18
   18 CONTINUE
C     REMOVE THE AREA FROM THE OLD CODE SO IT MATCHES THE
C     INCOMING IAREQ CODES, WHICH DONT HAVE AREAS ON THEM
      OLD = MOD(OLD,10000)
      IACONVT(IFILE,1) = OLD
      IACONVT(IFILE,2) = NEW
   20 CONTINUE
C     TAKE THE IAREQ CARDS OFF OF THE READER
   21 READ 101, (IAREQ1(I),I=1,10)
      IF (EOF,60) 60,25
C     SEARCH IACONVT FOR NEW CODE
   25 DO 30 JFILE=1,1500
      IF (IAREQ1(1) .EQ. IACONVT(JFILE,1)) 32,30
   30 CONTINUE
C     NO CODE FOUND - STORE IN ERROR LIST
      NO = 8H NO CODE
      WRITE (5,201) NO, (IAREQ1(I),I=1,10)
      GO TO 21
   32 CONTINUE
      IACONVT(JFILE,3) = 1
      IF (IAREQ1(2) .EQ. 0 .OR. IAREQ1(3) .EQ. 0) GO TO 43
      IF (IAREQ1(10) .EQ. 0) 43,33
   43 NO = 6060606060605460B
      WRITE (5,201) NO, (IAREQ1(I),I=1,10)
      GO TO 21
   33 CONTINUE
      IAREQ2(1) = IACONVT(JFILE,2)
C     CONVERT INTENSITY-DAYS TO HOURS-WEEKS
      IF (IAREQ1(3) .EQ. 1) 35,37
   35 IAREQ2(2) = IAREQ1(2)/10
      IF (IAREQ2(2) .LT. 1) IAREQ2(2)=1
      IAREQ2(3) = 1
      GO TO 50
   37 IF (IAREQ1(3) .LE. 5) 39,41
   39 IHRS = IAREQ1(2)/10
      IAREQ2(2) = IHRS * IAREQ1(3)
   41 IHRS = IAREQ1(2)/10
      DAYS = IAREQ1(3)
      WKS = DAYS/5. + .99
      IAREQ2(3) = WKS
      IAREQ2(2) = (IHRS * IAREQ1(3))/IAREQ2(3)
C     FINISH STUFFING THE NEW ARRAY
   50 IAREQ2(4) = IAREQ1(4)
      IAREQ2(5) = IAREQ1(5)
      K = IAREQ1(6) * 100
      IAREQ2(6) = K + IAREQ1(7)
      IF (IAREQ1(8) .EQ. 0) 52,54
   52 IAREQ2(7) = 0     $     GO TO 55
   54 K = IAREQ1(8) * 100
      IAREQ2(7) = K + IAREQ1(9)
   55 IAREQ2(8) = IAREQ1(10)
C     DONE WITH CONVERSION
C     PRINT AND PUNCH THE FILES
      PRINT 112, (IAREQ1(I),I=1,10), (IAREQ2(J),J=1,8)
      WRITE(62,114) (IAREQ2(J),J=1,8)
      KOUNT = KOUNT + 1
      GO TO 20
   60 CONTINUE
      PRINT 121, KOUNT
C     PRINT OUT ERROR FILE
      REWIND 5
      PRINT 116
      DO 65 I=1,1000
      READ (5,201) NO, (IAREQ1(J),J=1,10)
      IF (EOF,5) 66,64
   64 PRINT 118, NO, (IAREQ1(J),J=1,10)
   65 CONTINUE
   66 CONTINUE
      PRINT 123
      NOR = 0
      DO 70 I=1,IFILE
      IF (IACONVT(I,3) .EQ. 0) 72,70
   72 NOR = NOR + 1
      PRINT 125, (IACONVT(I,K),K=1,2)
   70 CONTINUE
      NET = IFILE - NOR
      PRINT 127, NOR, IFILE, KOUNT
      REWIND 3
      STOP
```

```
 91 PRINT 93     $     STOP
 93 FORMAT (10X,*** * * * ,*CONVERSION TAPE FILE SPLITTER NO GOOD*)
103 FORMAT (I8,1X,I7)
104 FORMAT (I10,1X,I6)
101 FORMAT (6X,I4,9I5)
112 FORMAT (10X,I4,9I8,10X,I6,2X,3(I2,1X),I3,2X,I3,2X,I3,2X,I2)
114 FORMAT (I6,8X,2I2, 6X,I2,I3,1X,I3,I3,4X,I2)
116 FORMAT (1H1,20X,*E R R O R  L I S T *,//)
118  FORMAT (10X,A8,2X,I10,1X,4(I2,1X),3(I3,1X),I2,1X,I2)
120  FORMAT (7X,*O L D  I A  C O D E *,17X,*N E W  I A  C O D E *,//)
 121 FORMAT (//,10X,*TOTAL IAREQ PUNCHED =*,I6)
 201 FORMAT (A8,10I10)
 123 FORMAT (1H1,20X,*THESE IA HAVE NO IAREQ REFERENCE THEM*,//)
 125 FORMAT (10X,2I10)
 127 FORMAT (////,10X,*NUMBER OF BUM IA IS *,I4,5X,* NET GOOD IA ARE *,
    1I4,10X,* WHICH HOPEFULLY IS CLOSE TO NET PUNCHED, WHICH IS *,I4)
     END
         SCOPE
LOAD
RUN,10,10000
```

EDSIM - 4

EDSIM - 4 serves many of the same purposes as EDSIM - 2. It was created to be more efficient and to provide better displays of information to decision-makers. It should be used in place of EDSIM - 2 for most practical purposes.

The basic processing cycle of EDSIM - 4, shown in Figure 1, consists of two phases. During the first phase students select the PC's (Performance Criteria) they wish to prepare for and they are scheduled for the IA's (Instructional Alternative) they are going to use for this preparation. During the second phase the simulated time is incremented and for each IA just completed students are post-tested in the related PC. The incremental time period of this cycle is one week.[1]

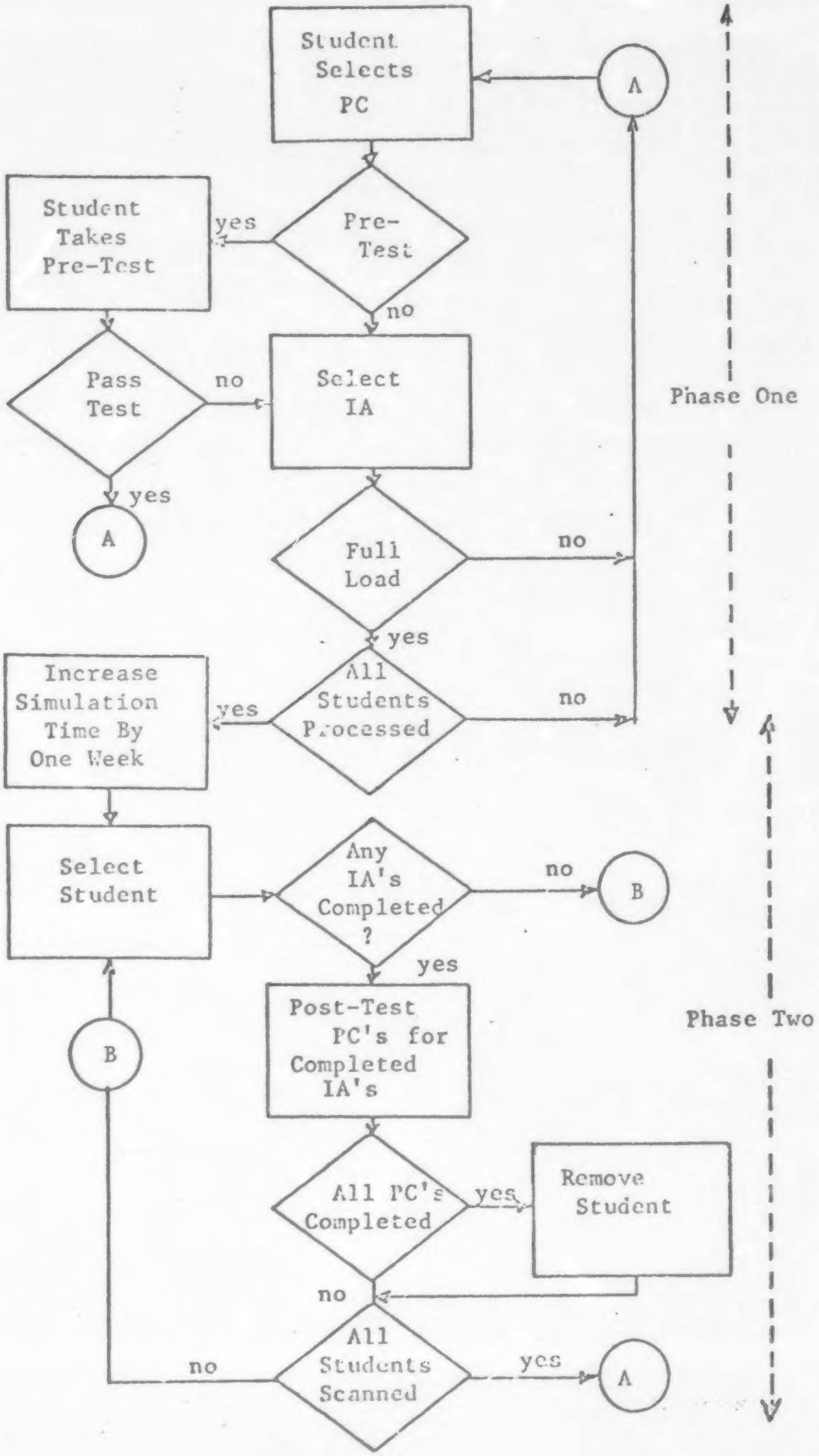
The following steps are performed for each simulated student during phase one. A PC that the student has not passed, is not currently engaged in, or scheduled to prepare for in a future time period is randomly selected using a discrete probability distribution.

---

[1] An incremental time period of one week was used in all of the studies performed with EDSIM - 4. By simple adjustments of the units of measure of the imput parameters, it would be possible to use any desired time period in a simulation.

FIGURE 1

Processing Cycle of EDSIM-4


Student Selects PC
A
Student Takes Pre-Test
yes
Pre-Test
no
Pass Test
no
Select IA
yes
A
Full Load
no
yes
Increase Simulation Time By One Week
yes
All Students Processed
no
Phase One
Select Student
Any IA's Completed ?
no
B
yes
Post-Test PC's for Completed IA's
B
Phase Two
All PC's Completed
yes
Remove Student
no
no
All Students Scanned
yes
A

then checked to determine whether it is offered during the current or future time periods. If it is not, an alternative IA is selected, or if no alternative can be scheduled then another PC is selected. If resources are not available to conduct the IA the student has just selected, he must select another PC. This first phase continues processing students sequentially until all students have a full load of instructional activities for the current time period.

During the second phase of the simulation cycle, the time clock is advanced one week and each student is examined to determine if he has completed any IA's during the previous week. If he has, he is post-tested using a discrete random variable to determine whether he passes the related PC, and his state with respect to the PC is updated. The state of the student is then checked to determine whether he has passed all PC's and if so, he is removed from the system. When all students are processed through the second phase, the first phase is initiated again. This cycle continues until the simulation is terminated. See documentation section for flowcharts.

All PC's and IA's are classified by event-types in a manner identical to the linear programming formulation. Each event-type requires the same use of resources.

When a student engages in a PC or IA, the required resources are debited. This is done in one of four ways. If the activity is a PC which is administered individually, the resources required are debited according to the requirements specified by the event-type. If the activity is a PC which is administered to a group of students, resources are debited on a prorated basis. For example, an examination which requires two hours of staff time to administer to thirty students would require 1/15 staff hours to administer to one student. That is not a restrictive assumption since many PC's are conducted on an individual basis. If the activity is an IA which is a prescheduled group activity, resources are debited at the beginning of each semester. If the activity is an IA which is conducted on an individual basis, resources are debited according to requirements specified by the event-type. This is done on a prorated basis according to the randomly generated student service time for the IA. Scheduling conflicts of resources or students within a single week are ignored.[2]

Service times for PC's and IA's are generated using normally distributed random variables with a specified mean and variance for each event-type. Noted that the service times are assumed to be continuous random variable and the model is updated on a discrete basis. The population of students is assumed to be homogeneous and these service times are not dependent on student attributes.

A number of assumptions concerning the structure and behavior of a performance criteria curriculum have been made in the development of EDSIM - 4. Since it is important that the user of this model be aware of these assumptions a complete list of them appears on page 24.

---

[2] Both students and resources are assumed to have a fixed number of hours available each week which may be devoted to their participation in the instructional system's activities. It is possible that a simulated student could be scheduled for two instructional activities which occur during the same time period during the week and likewise a resource such as a classroom might have a similar conflict. In this model these conflicts are assumed to have no significant effect on the model's response variables.

The structure of EDSIM - 4 is dependent upon seven data files which contain all relevant information concerning the configuration of the curriculum and the status of students and resources. A number of the elements of these files are constantly being updated as the simulated system moves through simulated time.

One additional element of the model's structure remains to be discussed, the model's data requirements and output reports.

Whenever possible the values of the parameters that describe student behavior and the operation of the instructional system are input into the model as data rather than being coded into the logic of the program. This was done to increase the model's flexibility and usefulness. The primary input data include:

1. The number of student arrivals at the beginning of each semester.
2. The probability of student selecting to pre-test for each PC.
3. The conditional probability of a student passing the related pre-test for each PC given he has decided to pre-test.
4. The probability of a student passing the related post-test for each PC.
5. The conditional probability of a student selecting an IA, given the student has selected a particular PC.
6. The mean service time and variance of this service time in hours for each IA.

7. The mean service time and variance of this service time in hours for each PC.
8. The subset of IA's which may be used to prepare the students for each PC.
9. The resources required to conduct each PC and IA.
10. The availability of each resource in hours per week.
11. A schedule for each IA indicating which weeks during the semester it will be offered.
12. The maximum allowable student load in hours per week.

The values of all of the above parameters must be carefully specified in order to utilize the simulation model. They can be obtained from operating data if an operating system exists,or if not subjective estimates can be used.

Except for the probabilities, this estimation procedure does not present any problems. Either the educator making the estimates will have data from which these values can be inferred or he will feel comfortable in making the estimates. However, when it comes to the probabilities, most educators are not used to thinking in these terms and usually do not have access to data from which inferences regarding the probabilities can be made. During this investigation it was found necessary to carefully instruct the individuals making these estimates regarding the exact type of information needed.

If an operating system exists, data from this system may be used to obtain estimates of the values of these parameters. As previously discussed, this method assumes there is a transferability of data from one instructional system to another if the two systems have similar characteristics. This assertion may be true only to a very limited extent.

It would therefore be desirable to develop a methodology which would enable an investigator to make inferences regarding the transferability of values of system parameters from one instructional system to another and to show the relevance of data from one set of students to another set of students participating in a performance criteria curriculum.

This type of approach in attempting to make inferences regarding the behavioral characteristics of one group of students to another would require stratification of the characteristics of the students. For example, to make an inference regarding the service time for a specific task that applies to a particular set of students based on data secured elsewhere,it might be necessary to classify students by age, IQ, and region of the country.

In order for the models developed during this investigation to come into widespread use, it would be necessary for this type of methodology to be available. For any particular application of the models, the data required for estimating system parameters are usually not available and must be collected by one of the procedures outlined above. This is a very time consuming and expensive process, hence the necessity for having this methodology.

The reports generated by EDSIM - 4 include:

1. Histograms indicating, by week, the mean, minimum, and maximum student load in hours per week.

2. Histograms indicating the number of students in the system by week and number of students completing all PC's each week.

3. A histogram of student loads in hours per week.

4. A histogram of student completion times in weeks.

5. A histogram for each resource showing its usage in hours per week.

6. The average weekly resources use and standard deviation of this use for each resource.

7. The total number of resource hours used during the period of the simulation and the total cost of this usage for each resource.

The majority of EDSIM - 4 is programmed in FORTRAN IV with a few subroutines programmed in COMPASS, the assembly language for the CDC 3600. One version of the model will handle up to 200 students, 50 PC's, 100 IA's, and 50 event-types. This corresponds to one pedagogic component of METEP. For the simulation of a performance criteria curriculum with 200 students, 27 PC's, 50 IA's and 15 event-types, the program runs about three minutes per simulated semester and requires about 32K of computer memory on the CDC 3600.

LANGUAGE ARTS COMPONENT SPECIFICATIONS

METEP, the instructional system from which much of the data used in this study was obtained, was used for two reasons: (1) the system was close at hand with design personnel readily available for consultation; and (2) since it is currently in its third year of design, program specifications could be obtained.

The language arts component of METEP is one of the twelve pedagogic components in which a student must show competency to complete his training in elementary school education. These pedagogic components are:

1. human relations
2. aesthetics
3. social studies
4. mathematics
5. pre-school education
6. media
7. teaching skills
8. language arts
9. science
10. foreign languages
11. evaluation skills
12. supervision

Detailed system specifications have been written only for the language arts, mathematics, science, and social studies components which were suitable for use by the methods of analysis developed during this investigation. Operating data were available from the language arts component and were used to develop the utility functions needed in the linear programming model, to estimate service times for IA and PC, and to estimate probabilities used in the simulation model.

The purpose of the language arts component is to develop skills and knowledge in the methods of teaching reading and language arts. This program of study consists of twenty-seven PC's covering a wide range of topics from the theory of beginning reading to IQ testing. A total of 261 instructional alternatives are incorporated into the program.

It was necessary to classify the PC's into five categories or event types for use in the linear programming and simulation models. These classifications are:

1. Demonstration of a concept or ability. For example, demonstrate the use of three reading machines.
2. Discussion with a staff member. For example, describe three different ways of presenting a story to a class of twenty children.
3. Participation in an activity, where the student is required to engage only in an instructional alternative. This type of PC is said to have an expressive objective. For example, participate in a discussion of the theories of beginning reading.
4. Writing of a short report. For example, evaluate in writing three elementary school reading texts.
5. Writing of a long report. For example, write a paper which defends the phonics approach to teaching reading.

For each of the twenty-seven PC's, students have the choice of ten IA event types which they can use to secure the knowledge and skills necessary to pass the PC. An attempt has been made in the design of this pedagogic component to include an IA of each type for every PC except in the case of PC's which have expressive objectives. The ten IA types are:

1. Lectures. Class lectures scheduled in advance, which include time for questions and discussion.

2. Lectures on audio tape. They may be listened to at any time in learning laboratory.

3. Library folder of readings. Each PC has a folder containing articles, excerpts from books, and samples of helpful materials.

4. Library browsing. Suggested reading lists have been compiled on materials related to each PC. Student may also browse and select his own readings independent of any suggestions.

5. Informal discussion with staff. Staff members will be available on a scheduled basis to give advice and information.

6. Informal discussion with others. Roommates, classroom teachers, former students, other students, and children are among the people who can be helpful to the student in learning how to prepare for the PC.

7. Audio-visual materials. Useful TV tapes, records, film strips, slides, and other audio visual presentations are available for each PC in the language arts learning laboratory.

8. Classroom observation. Elementary classes are available for observation by students.

9. Practice with elementary school class. Students can arrange for short practice sessions with elementary school classes. Often this is done as part of the student's student teaching.

10. Other activity. Student may select any other activity he feels would be helpful in preparing for the PC.

The instructional resources required to conduct the METEP pedagogic components can be classified into twelve categories:

1. Professional staff members
2. Graduate teaching assistants
3. Laboratory technicians required to support learning laboratories
4. Classrooms

5. Learning laboratories
6. Library
7. Micro-teaching laboratory (not used in language arts component)
8. Elementary classrooms and pupils
9. Elementary pupils for micro-teaching
10. Auditorium
11. Office space for consultations
12. Elementary classroom observation space

These resource categories are used in both the linear programming and the simulation models. The reports from each model display their utilization by category.

To facilitate modeling of the language arts component, estimates were made of the amount of each resource utilized when each IA and PC is conducted. These estimates were classified into the seventeen event types shown in Tables 1 and 2, Table 1 shows the resources required in hours of resource use to conduct each event type. The resource codes correspond to the numbers on the above resources. Table 2 shows the mean and variance of the preparation and contact time, the number of weeks the activity lasts, and the three probabilities associated with each PC. All lectures, seminars, and other group activities are pre-scheduled during the semester.

TABLE 1

RESOURCE REQUIREMENTS FOR EVENT TYPES
(ALL ENTRIES IN HOURS OF USE)

	RESOURCE CODES											
EVENT TYPE	1	2	3	4	5	6	7	8	9	10	11	12
1 LECTURE	3.00	.25	.02	.00	.00	.00	.00	.00	.00	1.00	.00	.00
2 AUDIO TAPE	.01	.01	.03	.00	1.00	.00	.00	.00	.00	.00	.00	.00
3 FOLDERS	.01	.01	.00	.00	.00	1.00	.00	.00	.00	.00	.00	.00
4 BROWSING	.01	.00	.00	.00	.00	1.00	.00	.00	.00	.00	.00	.00
5 DISC. STAFF	.25	.25	.00	.00	.00	.00	.00	.00	.00	.00	.50	.00
6 DISC. OTHERS	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
7 A.V. MATL.	.01	.00	.02	.00	1.00	.00	.00	.00	.00	.00	.00	.00
8 OBSERVATION	.00	.00	.00	.00	.00	.00	.00	1.00	.00	.00	.00	1.00
9 PRACTICE	.00	.01	.00	.00	.00	.00	.00	1.00	.00	.00	.00	.00
10 EXPRESSIVE IA	.00	.00	1.00	1.00	.00	.00	.00	.00	.00	.00	.00	.00
11 OTHER	.01	.02	.01	.00	.02	.02	.00	.02	.00	.00	.02	.00
12 TOUR OF LIBR	.50	.50	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
13 DEMONSTRAION	.00	.20	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
14 DISCUSSION PC	.00	.25	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
15 EXPRESSIVE PC	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
16 REPORT 1	.00	.25	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
17 REPORT 2	.00	.30	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

NOTE - SEE PRECEDING PAGE FOR MEANING OF CODES

TABLE 2

EVENT TYPE PARAMETERS

EVENT TYPE CODE	A	B	C	D	E	F	G	H	I
1	.00	.00	1.00	.00	1.00	40.	.00	.00	.00
2	.00	.00	1.00	.45	1.00	1.00	.00	.00	.00
3	.00	.00	1.00	.50	1.00	1.00	.00	.00	.00
4	.00	.00	1.00	1.10	1.00	1.00	.00	.00	.00
5	.00	.00	.50	1.00	1.00	1.00	.00	.00	.00
6	.00	.00	.66	.50	1.00	1.00	.00	.00	.00
7	.00	.00	.30	.25	.00	.00	.00	.00	.00
8	.00	.00	2.50	2.00	.00	.00	.00	.00	.00
9	.00	.00	1.00	.25	.00	.00	.00	.00	.00
10	.00	.00	1.00	.00	1.00	1.00	.00	.00	.00
11	2.50	2.50	.00	.00	1.00	1.00	.00	.00	.00
12	.00	.00	1.00	.00	1.00	15.0	.00	.00	.00
13	.00	.00	.50	.75	1.00	1.00	.10	.80	.90
14	.00	.00	1.25	.50	1.00	1.00	.07	.60	.80
15	.00	.00	1.65	.33	1.00	1.00	.00	.00	1.00
16	.00	.00	2.50	1.00	1.00	1.00	.05	.80	.85
17	.00	.00	1.66	1.00	1.00	1.00	.02	.70	.80

LEGEND
A = MEAN PREPARATION TIME IN HOURS
B = VARIANCE
C = MEAN CONTACT TIME IN HOURS
D = VARIANCE
E = NUMBER OF WEEKS ACTIVITY LASTS
F = AVERAGE NUMBER OF STUDENTS (MAX. IF 1.0)
G = PROBABILITY OF PRE-TESTING
H = PROBABILITY OF PASSING PRE-TEST
I = PROBABILITY OF PASSING POST-TEST

DATA EDIT AND DISPLAY PROGRAMS

The first program, EDIT[1], serves three purposes: (1) it edits for consistency the PCIA file which contains the IA's which may be used to prepare for each PC, and IA pointer file which contains the event type for each IA; (2) it displays the contents of these files; and (3) it assigns sequential codes to PC and IA, and then it punches out new data decks for input into EDSIM - 4. The second program, DISPLAY, is set up as a subroutine which must be inserted in the input routines of EDSIM - 4. It computes the longest and shortest completion times, the average service times for IA's and PC's, and the percentage of IA's and PC's requiring various resources.

EDIT requires two input decks each followed by an end-of-file card. The first data deck is composed of PCIA cards and the second of IA pointer cards. The format of the IA pointer cards appears on page 36 and that of the PCIA cards in Table 3.

To use DISPLAY, place a CALL to it in the main program of the input routines of EDSIM - 4 as shown in the program listings, and incorporate its source or object deck in EDSIM - 4's program decks at run time. The program will select from the data files the parameters necessary to generate its reports. Table 2.1 shows representative output from this routine.

---

[1] EDIT is an extension and refinement of a program originally coded by researchers in the School of Business of the University of Massachusetts.

TABLE 2.1

TABULATION OF DESIGN PARAMETERS FOR FOUR METEP PEDAGOGIC COMPONENTS

	LANG. ARTS	MATH	SOC. STUDIES	SCIENCE
NUMBER OF IA	36	87	256	115
NUMBER OF PC	27	46	121	61
MAX. HOURS TO COMPLETE ALL IA	66.00	443.00	285.00	123.50
MIN. HOURS TO COMPLETE ALL IA	8.80	443.00	135.00	25.00
AVERAGE SERV. TIME FOR IA	1.15	4.86	3.46	.31
AVERAGE SERV. TIME FOR PC	1.46	.89	4.17	.21
HOURS REQUIRED TO COMPLETE PC	39.00	46.00	51.00	13.00
PERCENTAGE OF IA REQUIRING				
STAFF	00	78	85	00
GTA	96	44	81	88
LAB TECH	00	00	07	00
CLASSROOM	00	00	84	02
LABORATORY	00	00	00	65
LIBRARY	00	00	10	00
MICRO TEACH LAB	00	00	00	00
ELEM. CLASSROOM	00	00	00	00
ELEM. STUDENT	00	00	06	00
AUDITORIUM	00	00	00	00
OFFICE	00	00	00	00
OBSERVATION	00	00	00	00
PERCENTAGE OF PC REQUIRING				
STAFF	80	52	56	00
GTA	70	53	47	38
LAB TECH	40	00	29	46
CLASSROOM	01	47	49	33
LABORATORY	30	47	10	44
LIBRARY	30	00	30	27
MICRO TEACH LAB	00	00	00	00
ELEM CLASSROOM	30	00	00	00
ELEM STUDENT	00	00	08	00
AUDITORIUM	10	00	00	00
OFFICE	20	00	00	00
OBSERVATION	10	00	00	00

The following annotated program listings show the structure and logic of these two programs. They may be interpreted and the program modified by the interested reader.

Table 3

PCIA Card Format

Card Columns	Content
1 - 5	Number assigned to PC
6 - 10	Event type of PC
15 - 20	First IA which can be used to prepare for PC
21 - 25	Second IA
26 - 30	Third IA
31 - 35	Fourth IA
36 - 45	Not used
46 - 50	Probability of selecting first IA
51 - 55	Probability of selecting second IA
56 - 60	Probability of selecting third IA
61 - 65	Probability of selecting fourth IA

Note: Current version of EDIT is limited to four IA's per PC, but this may be modified by making minor changes in the program.

```
      C
      C
      C
              PROGRAM EDIT
      C
      C                                 THIS PROGRAM  EDITS THE PCIA FILE AND
      C                                 THE IA POINTER FILE FOR INPUT INTO EDSIM-4
      C
      C                                 WRITTEN BY TOM RICHARDS AND GEORGE WILLIAM
      C                                             SCH. OF ENGR.     SCH. OF BUS.
      C                                                 UNIVERSITY OF MASSACHUSETTS
      C
              COMMON /1/ IAPC(1300,6), IAREQ(1300), IACODE(4), IAFILE(1300) ,
             1  IPCX(1300)   ,IAT(400,10),IAR(20)   ,IPROB(10)
              COMMON /2/ KOUNT1,IACNT
              REWIND 1
              REWIND 2
              DO 5 I=1,1500     $     DO 5 J=1,6
            5 IAPC(I,J) = 0
              KOUNT1 = 0      $   KK = 0     $     LINE = 0
              ICNT = 1
              PRINT 201
      C                                 READ IN PCIA FILE
           77 READ 101, IPC,ITYPE, (IACODE(I),I=1,4)    ,(IPROB(I),I=1,4)
              IF (EOF,60              ) 50,13
           13 WRITE(2), IPC, ITYPE,(IACODE(I),I=1,4) ,(IPROB(I),I=1,4)
              ISUM = 0     $     KOUNT1 = KOUNT1 +1
      C                                    CHECK FOR DUPLICATE PC NOS.
              IPCX (ICNT) = IPC
              IKT = ICNT - 1
              DO 700 I = 1, IKT
              IF(IPCX(I) .NE. IPC) GO TO 700
              PRINT 1020, IPC,IPCX(I),I,ICNT
              GO TO 77
          700 CONTINUE
              ICNT = ICNT + 1
              DO 15 J=1,4
              IF (IACODE(J) .EQ. 0)     GO TO 15
           17 ISUM = ISUM + 1
           15 CONTINUE
              LINE = LINE + 1
              IF (LINE .EQ. 60) 20,22
           20 PRINT 203      $     LINE = 1
           22 CONTINUE
              PRINT 205, IPC, (IACODE(I),I=1,ISUM )
      C                                  SET UP IA TO PC FILE
              DO 70 I=1,ISUM
              DO 71 J=1,KK
              IF (IACODE(I) .NE. IAPC(J,2))    GO TO 71
           75 IAINDX = IAPC(J,1)
              IF(IAINDX .GE.  6) GO TO 71
              IAINDX = IAINDX + 1
              IAPC(J,1) =    IAINDX
              IAPC(J,IAINDX) = IPC
              GO TO 70
           71 CONTINUE
              KK = KK + 1
              IAPC(KK,1) = 3
              IAPC(KK,3) = IPC
              IAPC(KK,2) = IACODE(I)
```

```
   70 CONTINUE
      GO TO 77
   50 CONTINUE
      ENDFILE 02
      PRINT 207, KOUNT1
      PRINT 213,KK
      PRINT 209
C                                   PRINT OUT REVERSE SORT OF PCIA FILE
      LINE = 0
      DO 55 J=1,KK
      LINE = LINE + 1
      IF (LINE .EQ. 60) 52,55
   52 PRINT 209     $    LINE = 1
   55 PRINT 211, J, (IAPC(J,K),K=2,6)
      IACNT = KK
C                                  COPY IAPC    INTO IA FILE
      DO 57 J=1,KK
   57 IAFILE(J)   = IAPC(J,2)
C                               READ IN IA REQ FILE
      ICNT2 = 0
      ICNT = 1
      IAREA1 = 01
      DO 60 JCNT =1,1300
      READ(60,1000) IANO,ITYPE,ISCHED
  601 IF (EOF,60) 62,602
  602 WRITE(01      ) IANO, ITYFE,ISCHED
      IAREQ(JCNT) = IANO
C                                        CHECK FOR DUPLICATE IA NOS.
      IPCX (ICNT) = IANO
      IKT = ICNT - 1
      DO 600 I = 1, IKT
      IF(IPCX(I) .NE.IANO) GO TO 600
      PRINT 1022,IANO,IPCX(I),I,ICNT
      GO TO 60
 1022 FORMAT(*              DUPLICATE IA NUMSERS IN DECK *  4I10)
  600 CONTINUE
      ICNT = ICNT + 1
   60 CONTINUE
C                                 CHECK IAREQ AGAINST IAFILE
   62 ITALLY = 0    $    JTALLY = 0
      KKK = JCNT  -1
      DO 67 I=1,KKK
      DO 65 J=1,KK
      IF (IAREQ(I) .EQ. IAFILE(J)) 67,65
   65 CONTINUE
      ITALLY = ITALLY + 1
      PRINT 215
      PRINT 217, IAREQ(I), I
   67 CONTINUE
C                                  CHECH IAFILE AGAINST IA REQ
      DO 85 J=1,KK     $    DO 84 I=1,KKK
      IF (IAFILE(J) .EQ. IAREQ(I)) 85,84
   84 CONTINUE
      JTALLY = JTALLY + 1
      PRINT 219
      PRINT 221, IAFILE(J), J
   85 CONTINUE
      NIAREQ = KKK - ITALLY
      NIAFILE = KK - JTALLY
      IF (NIAREQ .EQ. NIAFILE) 86,87
   86 PRINT 223, NIAREQ, NIAFILE
      GO TO 90
   87 PRINT 225, NIAREQ, NIAFILE
   90 ENDFILE 01
      ENDFILE 02
      REWIND 1
      REWIND 2
      CALL OUTP
       STOP
  101  FORMAT(  6I5     ,10X,4I5)
  102 FORMAT (6X,I4)
  201 FORMAT (10X,*PC CODE*,6X,*FIRST IA*,6X,*SECOND IA*,6X,
     1*THIRD IA*,6X,*FOURTH IA*,/)
  203 FORMAT (1H1,10X,*PC CODE*,6X,*FIRST IA*,6X,*SECOND IA*,6X,
     1*THIRD IA*,6X,*FOURTH IA*,/)
  205 FORMAT (10X,I8,4X,I10,4X,I10,5X,I10,5X,I10)
  207 FORMAT (//,10X,*TOTAL NUMBER OF PC =*,I5)
  209 FORMAT (1H1,25X,*IA CODE*,10X,*FIRST PC*,8X,*SECOND PC*,/)
  211 FORMAT (14X,I4,6X,I10,4(8X,I8))
  213 FORMAT (//,10X,*TOTAL NUMBER OF IA =*,I6)
  215 FORMAT (1H ,15X,*THE FOLLOWING CODED REQUIREMENTS DO NOT HAVE COD
     1S IN THE PC-IA FILE*,//)
  217 FORMAT (25X,I10,5X,*CARD POSITION IN THE IAREQ DECK*,I5)
  219 FORMAT (1H ,15X,*THE FOLLOWING PC-IA CODES DO NOT HAVE CODES IN T
     1E REQUIREMENTS FILE*,//)
  221 FORMAT (20X,I10,5X,*POSITION IN IAFILE*,I5)
  223 FORMAT (//,10X,*LENGTH OF NET IAFILE = IAREQ*,5X,2I10)
  225 FORMAT (//,10X,*LENGTH OF NET IAFILE NOT EQUAL TO IAREQ*,5X,2I10)
 1000 FORMAT(I10,I10,O10)
 1010 FORMAT(10X,15I8)
 1015 FORMAT(*    MORE THAN 400 EVENT TYPES FOUND * 3I10)
 1020 FORMAT(*          DUPLICATE PC NUMBERS IN DECK *  4I10)
 1501 FORMAT(1H1)
      END
C
C
      SUBROUTINE OUTP
      COMMON /1/ IAPC(1300,6), IAREQ(1300), IACODE(4), IAFILE(1300) ,
     1  IPCX(1300)   ,IAT(400,10),IAR(20)   ,IPROB(10)
      COMMON /2/ KOUNT1,IACNT
      DIMENSION IX(100)
      WRITE(61,1501)
      REWIND 1
      REWIND 2
      ITT = 0
       LINE = 0
   10 CONTINUE
      READ(01       ) IANO, ITYPE,ISCHED
      IF(EOF,01) 25,100
  100 IPT = 0
      ITT = ITT + 1
      IAREQ(ITT) = IANO
      DO 200 I =1,IACNT
      IF(IANO .NE. IAPC(I,2)) GO TO 200
      IADX = IAPC(I,1)
      DO 150 J = 3,IADX
      IPT = IPT + 1
  150 IX(IPT) = IAPC(I,J)
  200 CONTINUE
       LINE = LINE +1
      IF ((LINE/60)*60 - LINE .EQ.-1) WRITE(61,1004)
      WRITE(61,1001) ITT,IANO, ITYPE, ISCHED
C                                REMOVE C FROM COL. TO PUNCH
C     WRITE(62,1002)        ITT, ITYPE, ISCHED
      GO TO 10
C***********************   EDIT IA S ************************************
   25 ICNT = 0
       LINE = 0
   26 READ (2), IPC, ITYPE,(IACODE(I),I=1,4) ,(IPROB(I),I=1,4)
      IF(EOF,02)  800,900
  900 ICNT = ICNT + 1
      ISUM = 0
      DO 15 J=1,4
      IF (IACODE(J) .EQ. 0)    GO TO 15
   17 ISUM = ISUM + 1
   15 CONTINUE
       DO 65 J = 1,20
   65  IAR(J) = 0
      DO 70 J=1,ISUM
      DO 71 I=1,ITT
      IF(IACODE(J) .NE. IAREQ(I)) GO TO 71
      IAR(J) = I
   71 CONTINUE
   70 CONTINUE
       LINE = LINE +1
       IF ((LINE/60)*60 - LINE .EQ.-1) WRITE(61,1005)
      WRITE(61,1060)ICNT,IPC ,ITYPE,(IACODE(I),IAR(I), I=1,ISUM)
       WRITE(61,1061) (IPROB(I),I=1,ISUM)
C                                REMOVE C FROM COL. TO PUNCH
C      WRITE(62,1003) ICNT,ITYPE, (IAR(I),I=1,ISUM)
C      WRITE(62,1003) ICNT,ICNT,(IPROB(I) ,I=1,ISUM)
      GO TO 26
  800 RETURN
 1000 FORMAT(6X,I4,4X,2I2,6X,I2,I3,2(1X,I1,I2),3X,I2)
 1001  FORMAT(10X,3I10,5X,O10)
 1002  FORMAT(2I10,5X,O5)
 1003  FORMAT(10I5)
 1004  FORMAT(1H1,///,10X, 4X,* IA NUMBER     EVENT*10X,* SCHEDULE */
     1 28X, *TYPE*)
 1005  FORMAT(1H1,3X,*  PC     TYPE     IA     IA     IA     IA     IA
     1          PROBABILITIES*/ 4X,*NUMBER   CODE NUMBER   CODE NUMBER
     2 CODE NUMBER*4X* OF SELECTION *)
 1017 FORMAT(1H ,10X,60X,4I15)
 1500 FORMAT(1H )
 1501 FORMAT(1H1)
 1502 FORMAT(1H , ///  )
 1060 FORMAT(1H*,1017)
 1061  FORMAT(1H ,72X, 4I5)
      END
C
```

```
C
C
       SUBROUTINE DISPLAY
COMMON
       COMMON/12/ ALOAD.ICOHORT.COHORT.STAR(20)
       DIMENSION TT(12)  .AA(3)  .SS(12)
       AVPC = 0.0
       AVIA = 0.0
       ANOIA = 0.0
       IFLG2 =0  $ GTON = 0   $  GTOM = 0.
       TOTPC = 0.0
      WRITE(61.1501)
      WRITE(61.1502)
       DO 500 IBLK =1.NBLK
C                                  PICK PERFORMANCE CRITERIA
       ITYPE = PCPT(IBLK)
       DAYS = TYPE(ITYPE.3)
       HRS = TYPE(ITYPE.2)
       HRS = TYPE(ITYPE.2)    +TYPE(ITYPE.1)
       AVPC = AVPC + HRS
       PCTIM  = HRS * DAYS
       DO 100 J =5,16
       IF(TYPE(ITYPE.J) .LE. 0.0) GO TO 100.
       TT(J-4) = TT(J-4) + 1.0
  100  CONTINUE
      TOTMIN = 0.0
      TOTMAX = 0.0
      AMIN = 9999999.9
      AMAX = -999999.9
       ICNT = PCIA(IBLK.1)
       N = ICNT + 1
       DO 300 IP = 2.N
C                                  PICK INSTRUCTIONAL ALTERNATIVE
       IACT = PCIA(IBLK.IP)  .AND. 00000000777777778
       IF(IACT .LE. 0 .OR. IACT .GT. NOIA ) GO TO 7778
       ITA = IAPT(IACT)
       IX = (   ITA     .AND. 7777777700000000B)
       ITYPE = (  ITA         .AND. 0000000077770000B)
       ITYPE = ITYPE/2**12
       IF(ITYPE .LE. 0 .OR. ITYPE  .GT.  NIAR) GO TO 7779
       HRS = TYPE(ITYPE.2)
       HRS = TYPE(ITYPE.2)    +TYPE(ITYPE.1)
       AVIA = AVIA + HRS
       ANOIA = ANOIA + 1.0
       DO 150 J =5,16
       IF(TYPE(ITYPE.J) .LE. 0.0) GO TO 150
       SS(J-4) = SS(J-4) + 1.0
  150  CONTINUE
       DAYS = TYPE(ITYPE.3)
       IF(DAYS .GT. 0.0) GO TO 290
       WRITE(61.1000) DAYS. ITYPE
       GO TO 300
  290  TIM  = HRS * DAYS
      IF(TIM  .LT. AMIN) AMIN = TIM
      IF(TIM  .GT. AMAX) AMAX = TIM
  300   CONTINUE
       IF( AMIN .EQ.  9999999.9 ) AMIN = 0.0
       IF( AMAX .EQ.  -999999.9 ) AMAX = 0.0
      TOTMIN = TOTMIN + AMIN
      TOTMAX = TOTMAX + AMAX
```

5-21

```
      TOTPC = TOTPC     + PCTIM
      GTON = GTON + AMIN
     GTOM = GTOM + AMAX
     WRITE(61,1012) IBLK  . TOTMIN. TOTMAX   .PCTIM
  500 CONTINUE
C                                     PRINT REPORTS
  600 WRITE(51.1013) GTON. GTOM .TOTPC
      A1 = GTON /ALOAD
      A2= GTOM/ALOAD
      A3 = TOTPC/ ALOAD
      WRITE(61.1014) A1.A2.A3.ALOAD
C                                     COMPUTE AVERAGE LENGTH OF IA AND PC
      WRITE(61.1001) AVIA.AVPC
      X = NBLK
      AVPC = AVPC/X
      AVIA = AVIA/ANOIA
      WRITE(61.1001) AVIA.AVPC
      WRITE(61.1005)
      DO 705 I =1.12
      K = I + 4
      TT(I) = TT(I)/X
  705 WRITE(61.1002) K. TT(I)
      WRITE(61.1005)
      DO 716 I =1.12
      K = I + 4
      SS(I) = SS(I)/ANCIA
  716 WRITE(61.1002) K. SS(I)
     STOP
 1000 FORMAT(20X.*ERRCR IN DATA - WEEKS LE 0.0 IN TYPE FILE *.2I10)
 1001 FORMAT(1H1.10X.*MEAN LENGTH OF INSTRUCTIONAL ALTERNATIVES IS*
    1 F10.2///10X.*MEAN LENGTH OF PERFORMANCE CRITERIA IS      *F10.2)
 1002 FORMAT(1H0.5X.  I10.F20.3)
 1003 FORMAT(7I10.F10.2)
 1004 FORMAT(1H0. 10I10)
 1005 FORMAT(1H1.10X.*RESOURCE                 PERCENT    *
    1/8X. *FILE POSTION         OF IA REQUIRING *//)
                                                                 M
 1012 FORMAT( 1H . 5X. I10.  3F20.2)
 1013 FORMAT(1H .9X.12HGRAND TOTALS   .F14.2.2F20.2// )
 1014 FORMAT(1H0.9X.*WEEKS * 3F15.2.* AT * F7.2. *  HRS PER WEEK * //)
 1501 FORMAT(1H1)
 1502 FORMAT(22X.* TABLE      */// 15X.*  MINIMUM AND MAXIMUM PATHS  *
    3 / 15X.*        (IN HRS   )    *
    1/// 10X. *  AREA * 10X. *MINIMUM TIME * 10X. *MAXIMUM TIME *//)
 7778 WRITE(61.1015) IACT. NOIA
 1015 FORMAT(10X.///*BAD INPUT DATA. I.A. NUMBER * I10.*  IS TOO LARGE
    1     THE NUMBER OF I.A. IN SYSTEM IS * I10)
      RETURN
 7779 WRITE(61.1016)     ITYPE. NIAR
 1016 FORMAT(10X.///* BAD INPUT DATA. EVENT TYPE* I10 *  LARGER THAN*
    1 *NO. OF EVENT TYPES * I10)
      STOP
     END
C
C
C
C
```

EDSIM - 4 DOCUMENTATION

Program Structure

This section contains complete documentation for EDSIM - 4, including assumptions used in program development, flowcharts, file structure, operating instructions, sample input, sample output, and program listings.

The approach used in this investigation was to determine the individual student's effect on the simulated system's resources and to record each student's progress through the system. To increase the usefulness of the model, what is normally a single student entity can be defined as representing a cohort or group of students. Each student within a cohort effects resources in the same manner or degree as every other as they simultaneously progress through the instructional system along identical paths.

EDSIM - 4 has two attributes which Conway[1] (8, P. 95) feels should be incorporated into all simulation models: flexibility in design and modular construction. This type of construction enhances the capability of easily making changes in the model which reflect changes in decision rules and assumptions regarding system operation. The program is composed of a number of subroutines, each performing a single processing function. Complete subroutines may be replaced, or their components may be modified without disturbing the rest of the model.

---

[1] Conway, R.W., "Some Tactical Problems in Digital Simulations", Management Science, Vol. 6 (1959), 92-110.

EDSIM-4 consists of three basic programs: an input program which edits and displays input parameters, the main simulation program, and a report generator. Figure 4 shows how these programs are linked together using magnetic tapes for intermediate output and Figure 5 contains detailed flowcharts of the main simulation program.

Numerous assumptions were made in the construction of EDSIM-4. The following list contains the specific assumptions which should be understood before EDSIM - 4 is used. The prospective user should also carefully review the flowcharts for EDSIM - 4.

1. Potential scheduling conflicts within the incremental time period of one week are ignored when scheduling both resources and individual students.
2. When a student selects a PC, a uniform probability distribution is used to determine in which PC he will engage. If he has already engaged in the PC selected, a sequential scan over all PC's is performed to select a PC the student has not completed.
3. Whenever a student requests to engage in a PC, it is conducted regardless of resource availability.
4. If a PC requires a group of students for its administration, the PC is conducted individually with the resources prorated.
5. If a student fails a PC, his probability of passing the PC remains constant on all succeeding trials.
6. Students are processed sequentially in the order they appear in the student file.
7. A student will not preschedule any IA's during a week for which his projected load for that week exceeds the maximum allowable load.
8. A student will not engage in any additional IA's during the current week if his current load is greater than the maximum allowable load.

FIGURE 4

Block Flowchart of Interface Between EDSIM-4 Programs

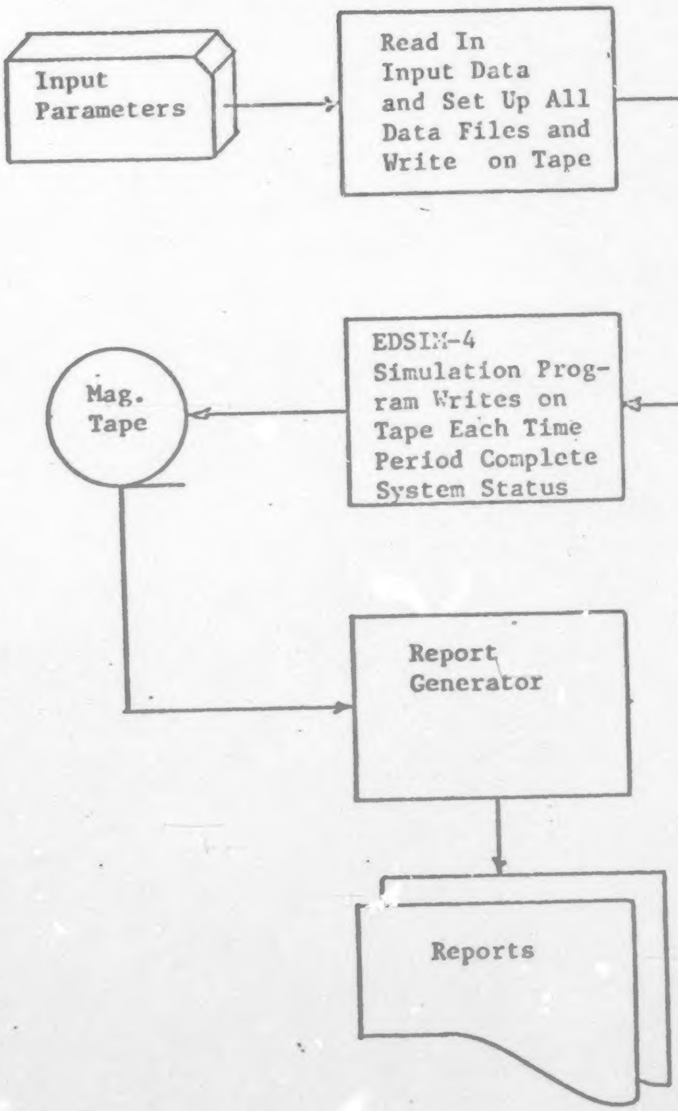

Input Parameters
Read In Input Data and Set Up All Data Files and Write on Tape
Mag. Tape
EDSIM-4 Simulation Program Writes on Tape Each Time Period Complete System Status
Mag. Tape
Report Generator
Reports

FIGURE 5

Flowchart of EDSIM-4

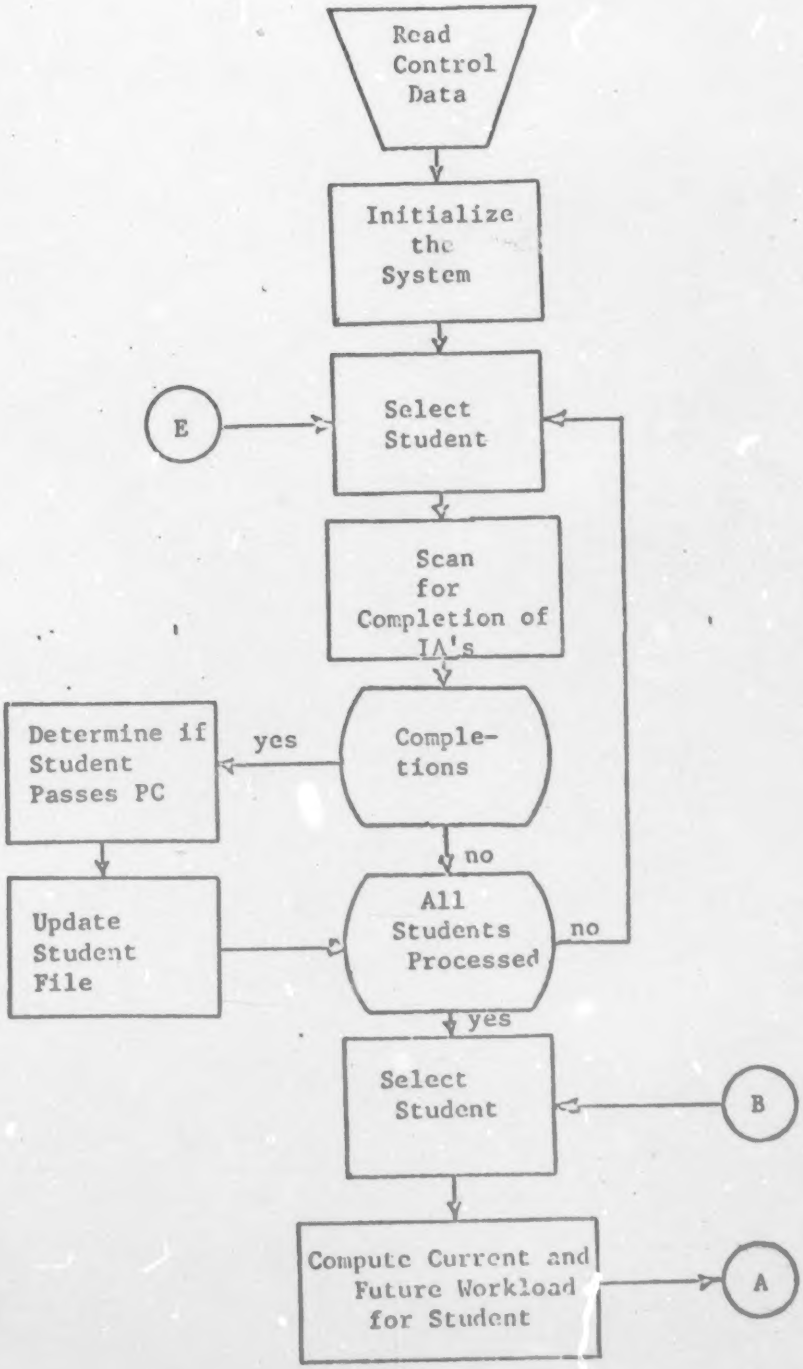

Read Control Data
Initialize the System
E
Select Student
Scan for Completion of IA's
Completions
yes
Determine if Student Passes PC
no
Update Student File
All Students Processed
no
yes
Select Student
B
Compute Current and Future Workload for Student
A

FIGURE 5 Continued

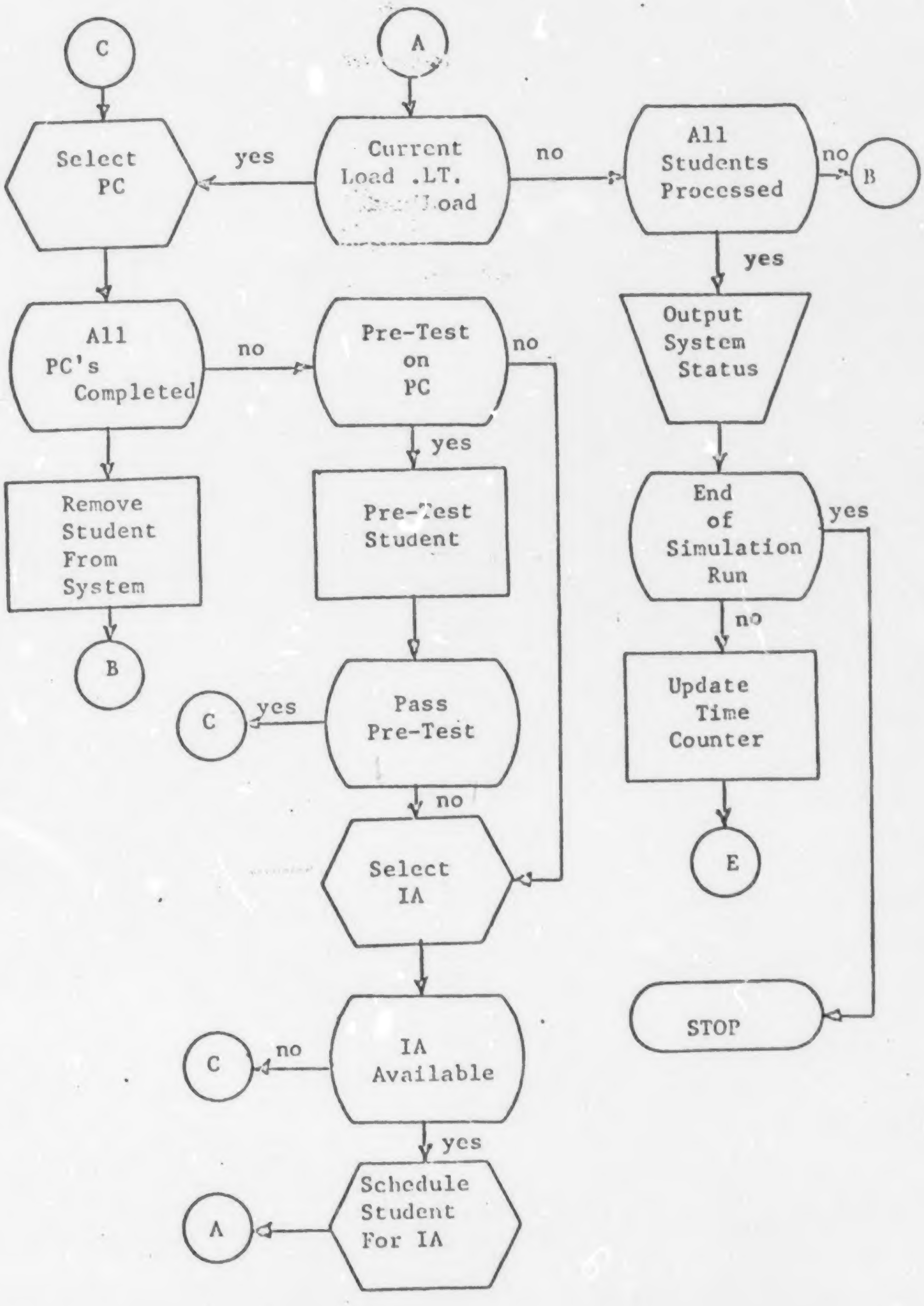

C
A
Select PC
yes
Current Load .LT. Max Load
no
All Students Processed
no
B
yes
All PC's Completed
no
Pre-Test on PC
no
Output System Status
yes
Remove Student From System
Pre-Test Student
End of Simulation Run
yes
B
no
C
yes
Pass Pre-Test
Update Time Counter
no
E
Select IA
STOP
C
no
IA Available
yes
A
Schedule Student For IA

FIGURE 5 Continued

Select PC Subroutine

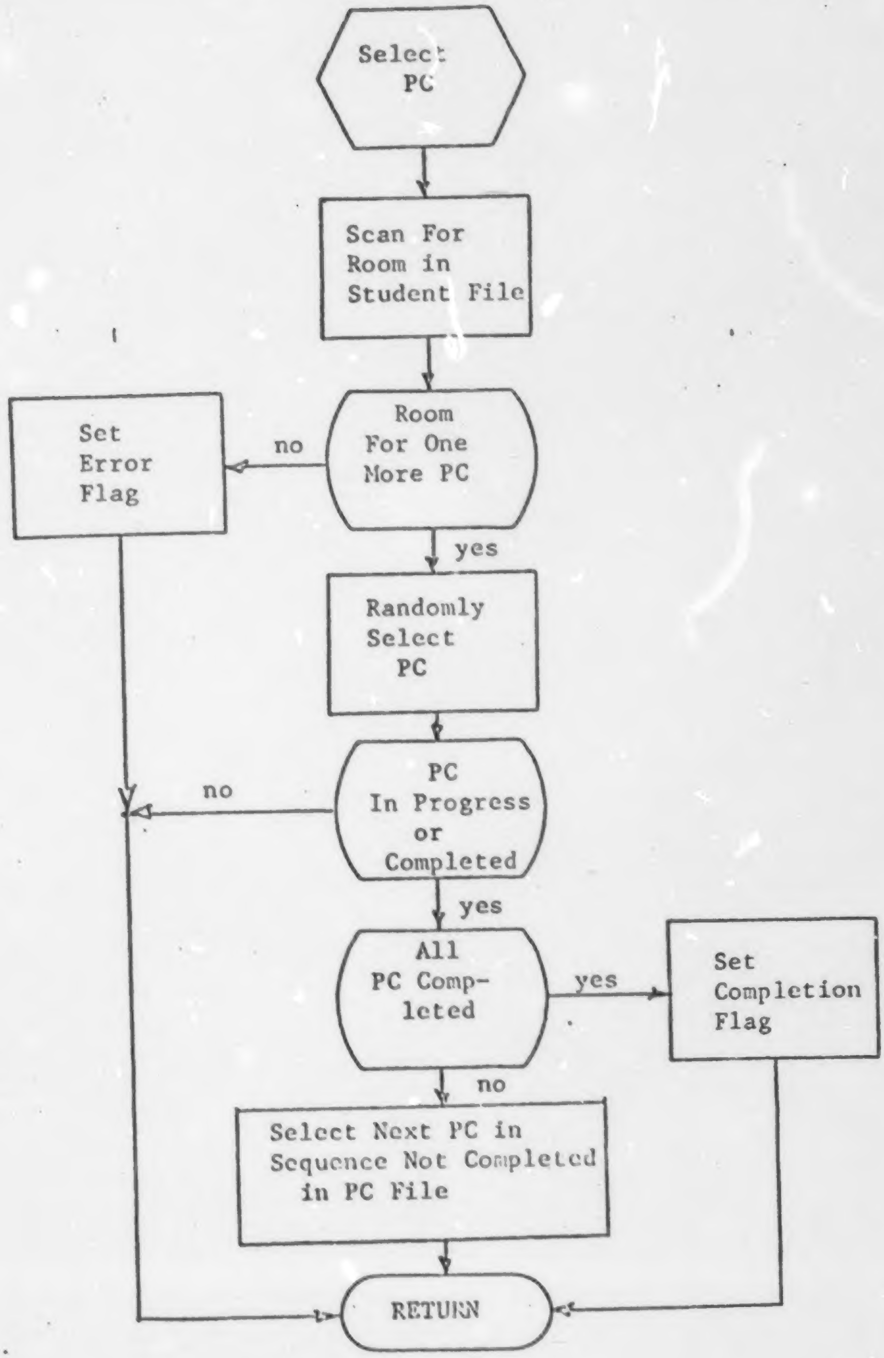

Select PC
Scan For Room in Student File
Room For One More PC
no
Set Error Flag
yes
Randomly Select PC
PC In Progress or Completed
no
yes
All PC Completed
yes
Set Completion Flag
no
Select Next PC in Sequence Not Completed in PC File
RETURN

FIGURE 5 Continued

Select IA Subroutine

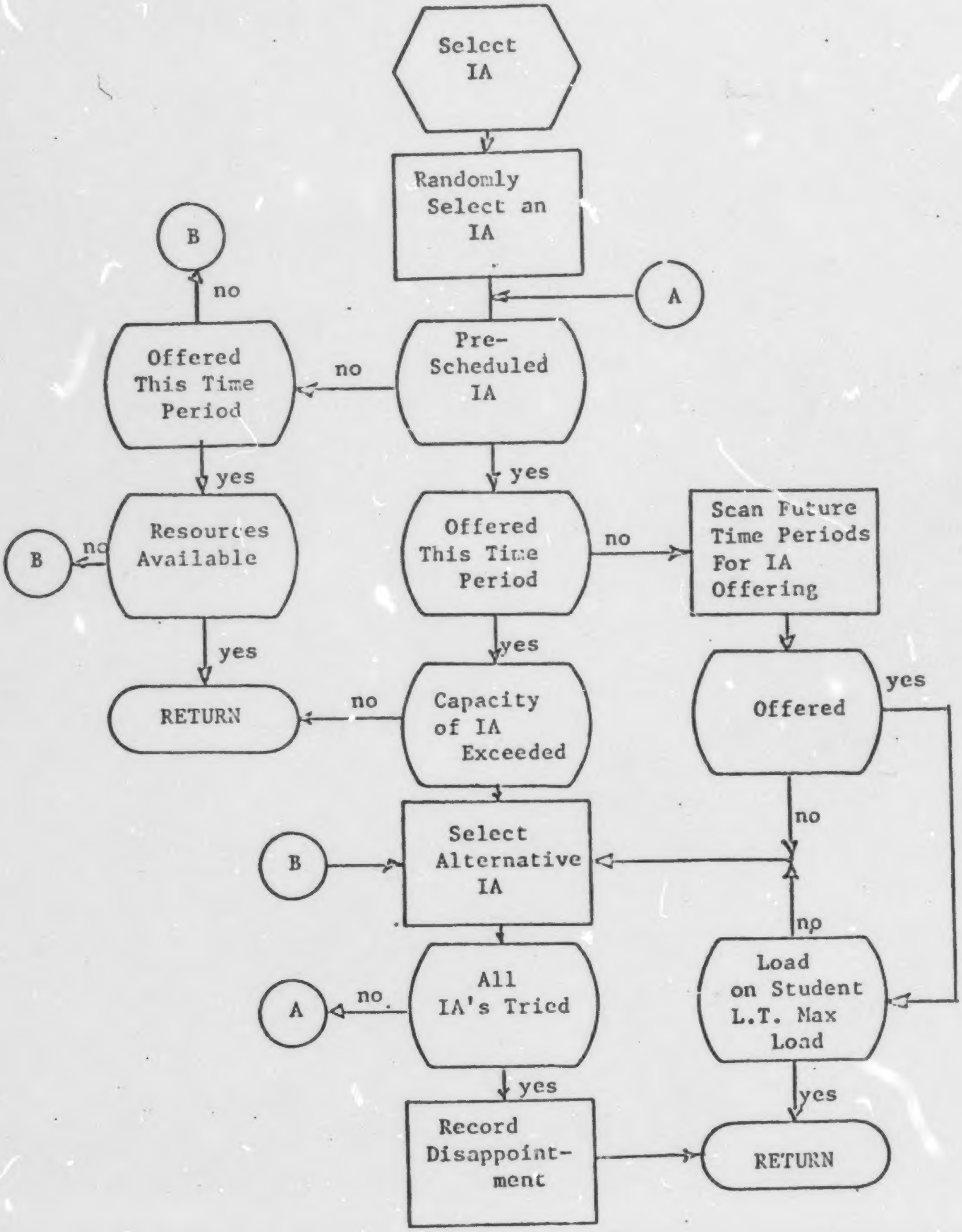
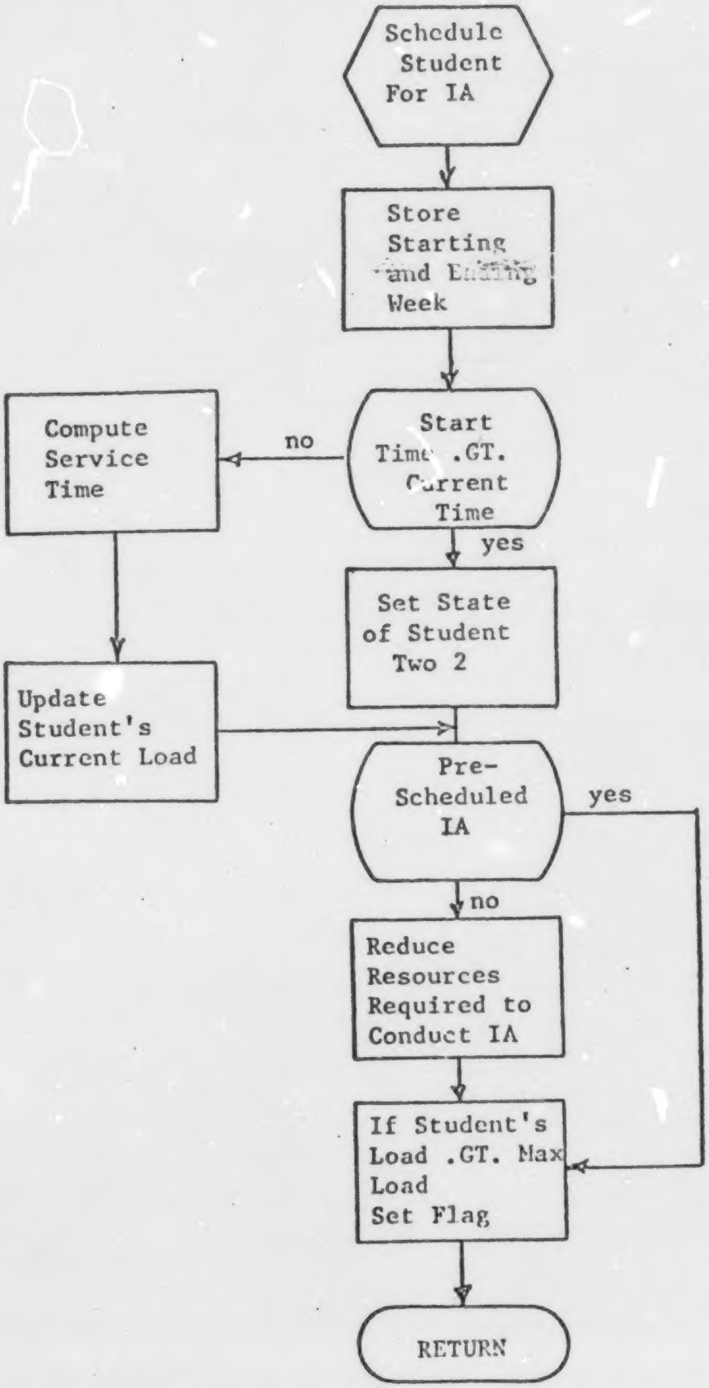

Select IA
Randomly Select an IA
B
no
A
Offered This Time Period
no
Pre-Scheduled IA
yes
yes
Resources Available
no
B
Offered This Time Period
no
Scan Future Time Periods For IA Offering
yes
yes
RETURN
no
Capacity of IA Exceeded
Offered
yes
no
B
Select Alternative IA
no
A
no
All IA's Tried
Load on Student L.T. Max Load
yes
yes
Record Disappointment
RETURN

FIGURE 5

Flowchart of Schedule Subroutine


Schedule Student For IA
Store Starting and Ending Week
Start Time .GT. Current Time
no
yes
Compute Service Time
Update Student's Current Load
Set State of Student Two 2
Pre-Scheduled IA
yes
no
Reduce Resources Required to Conduct IA
If Student's Load .GT. Max Load Set Flag
RETURN

9. If an IA is not offered during the current time period, the student will attempt to schedule the IA during a following week. It may be scheduled during any week up to the end of the semester.

10. If an IA is not offered during the current week or a future week, the student will select an alternative IA.

11. A student may engage in the same IA several times to prepare for the same or different PC. No record is kept of IA's a student has engaged in.

12. If resources are not available to conduct an IA, a student will not engage in the IA.

13. Student service times for prescheduled IA's are recomputed each week.

14. A student engages in only one IA before attempting to pass the related PC.

15. The amount of resource used to conduct a prescheduled IA is constant regardless of the number of students engaged in the activity.

EDSIM-4 File Structure

The structure of EDSIM-4 is dependent upon seven data files which contain information used by the program. Much of the data in these files is constantly being updated and modified.

1. The student file contains all pertinent information regarding each student. This information includes the IA's the student is currently engaged in, the PC's which he is preparing to meet, the starting week of any IA's he is scheduled to engage in during future weeks, the week during which he will complete service in any IA's he is engaging in, the student's current load in hours per week, and the semester he entered the instructional system.

2. The state file contains the state of each student with respect to each PC. A one is recorded if the student has passed the PC, a two is recorded if the student is currently engaged in an IA which prepares him for a particular PC.

3. The PCIA file contains the IA's which may be used to prepare a student for each PC and the probability for each of these IA's that it will be used, given a student has selected to prepare for a particular PC.

4. The IA pointer file contains a schedule for each IA indicating the weeks during the semester it will be offered, the event type associated with the IA, and the number of students engaged in the IA during the current time period.

5. The PC pointer file contains the event type associated with the PC.

6. The hours available file contains the number of resource hours available during each week of the simulation for each resource. The elements of this file are reduced as students engage in IA's and PC's.

7. The resource requirements file has entries which indicate the amount of each type of resource necessary to conduct each event. These entries are in hours per week.

Each data file used in the model has one COMMON card which may be replaced if the size of the file is to be modified. The user is cautioned also to change the values of the variables which specify the maximum dimensions of each file in the input routines of EDSIM-4. These are used by the program to check indices before data is stored or retrieved. Several of the files have pointers which indicate the location of component parts of the file which must also be modified if the file size is changed. These parameters appear immediately after the dimension of the file in its COMMON card.

The detail structure of each of the above files follows:

1. Student File. FORTRAN name of file D(M,N) or FD(M,N)

One row per student and one 48 bit word per column.

Column	Content of Column
1	PC student is preparing to meet
2	IA student is engaging in to meet PC
3	First half word, starting week of IA Second half word, ending week of IA

The above sequence is repeated K times (the pointer ATRPT contains the value of 3K+1) then the following is recorded for each student:

Column	Content of Column
3K+1	Number of students in cohort
3K+2	Maximum student load in hours per week
3K+3	Current student load
3K+4	Entry week of student into system

2. State File. FORTRAN name of file STATE(M,N)

One row per student. Columns are packed 24 columns per 48 bit word. States are: 1 = completed PC, 2 = currently engaged in PC, and 0 = otherwise.

Column	Content of Column
1	State of student with respect to PC 1.
2	State of student with respect to PC 2.
3	State of student with respect to PC 3.
.	. . . .
.	. . . .
.	. . . .
n – 1	State of student with respect to PC n-1.
n	State of student with respect to PC n.

3. PCIA File. FORTRAN name of file PCIA(M,N)

One row per PC. Row number corresponds to PC number. One 48 bit word per column. Two digit integer probabilities are used.

Column	Content of Column
1	Number of IA associated with PC number 1
2	First half word, probability of selecting IA Second half word, IA number
3	First half word, probability of selecting second IA Second half word, second IA number

Up to N-1 IA may be associated with each PC.

4. Resource Hours Available File. FORTRAN name of file HOURS(M,N)

One row for each week of semester. One 48 bit word per column.

Column	Content of Column
1-4	Not used
5	Staff hours available per week
6	Graduate teaching assistant hours available per week
7	Laboratory technician hours available per week
8	Classroom hours available per week
9	Laboratory hours available per week
10	Library hours available per week
11	Micro teaching hours available per week
12	Elementary classroom hours available per week
13	Elementary student hours available per week
14	Auditorium hours available per week
15	Office hours available per week
16	Observation hours available per week

Note: The particular types of resources specified by each column are a function of the system under study. The above specifications correspond to the headers used in the reports generated by EDSIM-4. These could be modified with ease.

5. Resource Requirements File. FORTRAN name of file TYPE(M,N)

One row for each event type. One computer word per column.

Column	Content of Column
1	Mean student preparation hours required for activity
2	Mean student contact hours required per week for activity
3	Length of activity in weeks
4	Maximum students per activity
5-16	Columns 5 through 16 contain the mean resource required for the twelve resource types appearing in the HOURS file
17	Probability of selecting to pre-test if event type is a PC
18	Probability of passing pre-test
19	Probability of passing post-test
20	Standard deviation of required preparation hours

21	Standard deviation of required contact hours
22	First bit not used Second bit is a one if event type is a pre-scheduled group activity Third bit is a one if event type has a fixed service time Fourth bit is a one if event type has an expressive objective which means student does not pre- or post-test

Note: The current version of FDSIM-4 used a normal distribution to generate service times. This subroutine could be replaced with any desired type of random variate generator. Up to four parameters can be input into this routine from columns 1, 2, 20, and 21.

6. IA Pointer File. FORTRAN name of file IAPT(N)

One row per IA and 12 or 24 bits per column.

Column	Content of Column
1	The first 24 bits of the 48 bit computer word indicate the schedule of the IA. A one in the nth position indicating the IA is offered during the nth week of the semester.
2	A twelve bit event type code
3	The number of students currently engaging in IA

7. PC Pointer File. FORTRAN name of file PCPT(N)

One row per PC and one word per row which contains the event type for the PC.

Operating Instructions

The program requires three magnetic tapes: tape unit 10 is used for dumping system status parameters which are used later by a report generator, and units 9 and 11 are used for temporarily storing the HOURS file. When the program is run on the CDC 3600, a system DEMAND card must be included in the system control cards.

Figure 6 shows the normal organization of the program decks. Three programs and two data decks are required. The input program sets up and initializes the data files used by the main program. The main program generates data on tape unit 10 which is used for input into the report generator. These programs may be run separately using tape unit 10 to transfer data from one program to another or they may be run as shown in Figure 6.

The format of the input data cards and a sample listing of these cards follow. (See Table 4).

1. Simulation Control Cards.

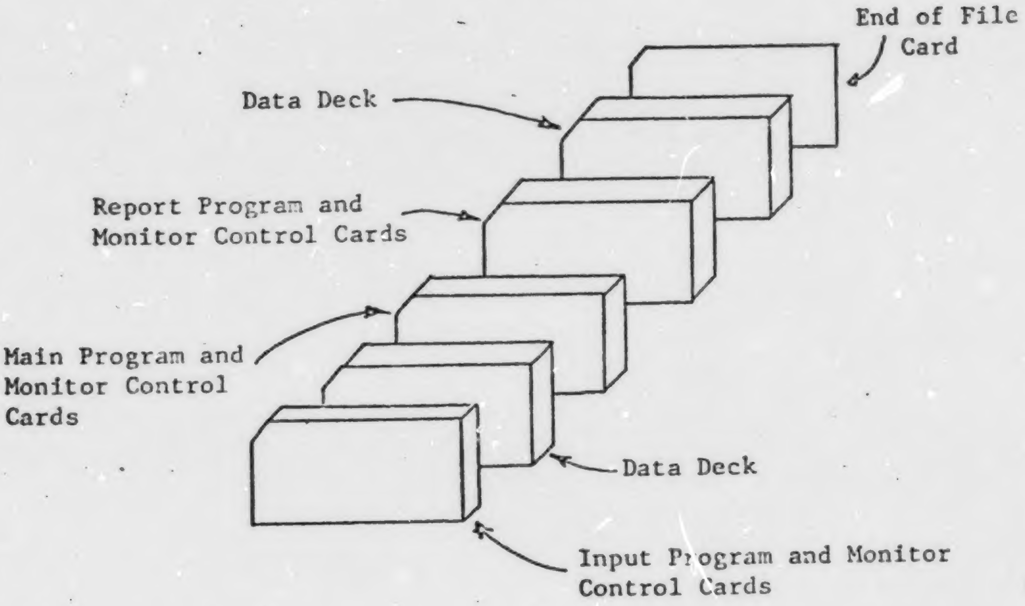
Card Number	Content of Card	Format
1	First dump flag. If zero system status is dumped on tape 10 each week. Second dump flag. If non-zero complete system status dumped on printer each simulated week.	2I10
2	Starting semester for simulation	50X,I10
3	Ending semester for simulation	50XI10
4	Case number for simulation run	50X,I10
5	Number of weeks per semester	50X,I10
6	Maximum student load in hours per week	50X,F10.0
7	Number of students per cohort	50X,I10
8	Number of weeks student will look ahead in selecting IA	50X,I10

2. PCIA Cards.

Two cards are required for each PC.

FIGURE 6

Normal Organization of Program Decks for Execution of EDSIM-4


End of File Card
Data Deck
Report Program and Monitor Control Cards
Main Program and Monitor Control Cards
Data Deck
Input Program and Monitor Control Cards

Card Number one. Format (1615)

Card Columns	Content
1 - 5	PC number
6 - 10	PC event type
11 - 15	First IA
16 - 20	Second IA
21 - 25	Third IA
.	.
.	.
.	.
75 - 80	Fourteenth IA

Card Number two. Format (15,5X,1415)

Card Columns	Content
1 - 5	PC number
6 - 10	Blank
11 - 15	Two digit probability of selecting first IA
16 - 20	Two digit probability of selecting second IA
.	.
.	.
.	.
75 - 80	Two digit probability of selecting fourteenth IA

Note: The maximum number of IA's per PC is dependent on the size of the PCIA file.

3. Instructional Alternative Cards.

One card is required for each IA. Format (2110,5X,016)

Card Columns	Content
1 - 10	IA number
11 - 20	Event type code
21 - 24	Not used
26 - 35	Schedule of IA. The octal representation of the binary number which indicates the weeks in which the IA occurs.

Examples of the last field are:

10000 = IA is offered during the first week of the semester

20000 = IA is offered during the second week of the semester

77777 = IA is offered every week of the semester

4. Event Type Cards.

Each event type requires two cards input in sequence.

Card number one. Format (I2,I3,15F5.0)

Card Columns	Content
1 - 2	01
3 - 5	Event type number
6 - 10	Mean student preparation hours
11 - 15	Mean student contact hours
16 - 20	Length of event type in weeks
21 - 25	Maximum students per event type
26 - 30	Staff hours required
31 - 35	GTA hours required
36 - 40	Laboratory hours required
41 - 45	Classroom hours required
46 - 50	Laboratory hours required
51 - 55	Library hours required
56 - 60	Micro teaching hours required
61 - 65	Elementary classroom hours
66 - 70	Elementary student hours
71 - 75	Auditorium hours required
76 - 80	Office hours required

Card number two. Format (I2,I3,15F5.0)

Card Columns	Content
1 - 2	02
3 - 5	Event type number
6 - 10	Observation hours required
11 - 15	Probability of selecting to pre-test if event is a PC
16 - 20	Probability of passing pre-test
21 - 25	Probability of passing post-test
26 - 30	Standard deviation of preparation hours
31 - 35	Standard deviation of contact hours

If needed, a third card may be input for each event type. Not all event types require this card.

Card number three. Format (I2,I3,5X,016)

Columns	Content of Field
1 - 2	03
3 - 5	Event number
6 - 10	Not used
11 - 26	Octal code for control bits (see resource requirements file for codes)

5. Resource Cards. Format (4A8,8X,I10,2F15.0)

These cards specify the amount of each resource which will be available during each week of a simulated semester.

Columns	Contents
1 - 40	Descriptive comments
41 - 50	Position in HOURS file (column number)
51 - 65	Hours contributed per week per unit of resource
66 - 80	Total units of resource available

6. Resource Cost Cards. Format (4A8,8X,I10,F15.0)

These cards specify the units costs of resources.

Columns	Contents
1 - 40	Descriptive comments
41 - 50	Position in HOURS file
51 - 65	Unit cost of one research hour

This appendix contains selected output reports from EDSIM-4. A complete set of reports was not included due to their volume. The reports included are representative of the type of information displayed by the report generator of the simulation model.

A complete listing of EDSIM-4 follows this sample output. These program listings are annotated to facilitate their interpretation. To reduce their length, COMMON statements have not been included, but a sample of the COMMON cards which should be included in each program or subroutine is included.

TABLE 4

SAMPLE INPUT FOR EDSIM-4

```
    ************************************
SOURCE OR OBJECT DECK FOR INPUT ROUTINES GOES HERE
PLUS APPROPRIATE MONITOR CONTROL CARDS
    ************************************

99999     0
 1 STARTING SEMESTER FOR SIMULATION RUN IS                              1
 2 ENDING SEMESTER FOR SIMULATION RUS IS                               52
 3 CASE NUMBER FOR SIMULATION RUN IS                                   25
 4 NUMBER OF WEEKS PER SEMESTER IS (MAX OF 15)                         15
 5 MAXIMUM LOAD IN HOURS PER WEEK IS                                   15
 6 NUMBER OF STUDENTS PER COHORT IS                                     1
100   100   100   100   100   100   100   100   100   100   100   100
 7 NUMBER OF WEEKS LOOK AHEAD IN SELECT ACTS                            8 1
  01    17     1     2     3     4     5     6     7     8     9    11
  01          45    14    06    17    04    04    00    00    04    06
  02    14    12     2     3     4     5     6     7     8     9    11
  02          15    00    07    11    19    04    00    00    00    44
  03    16    13     2     3     4     5     6     7     8     9    11
  03          49    05    18    07    05    05    02    02    05    02
  04    17    14     2     3     4     5     6     7     8     9    11
  04          53    02    14    10    02    02    00    03    07    07
  05    17    15     2     3     4     5     6     7     8     9    11
  05          62    02    04    04    02    12    00    02    12    00
  06    17    16     2     3     4     5     6     7     8     9    11
  06          50    04    21    07    00    01    00    00    07    10
  07    17    17     2     3     4     5     6     7     8     9    11
  07          47    00    32    00    03    08    00    00    05    05
  08    14    18     2     3     4     5     6     7     8     9    11
  08          49    00    22    13    07    00    00    00    07    02
  09    14    19     2     3     4     5     6     7     8     9    11
  09          38    04    20    14    06    04    00    00    10    04
  10    17    20     2     3     4     5     6     7     8     9    11
  10          39    00    24    15    00    04    00    00    18    00
  11    16    21     2     3     4     5     6     7     8     9    11
  11          30    03    25    00    06    16    00    00    06    14
  12    14    22    02    03    04    05    06    07    08    09    11
  12          28    14    25    08   003    08    00    00    00    14
  13    13    23     2     3     4     5     6     7     8     9    11
  13          52    00    03    00    06    00    00    00    33    06
  14    16    24     2     3     4     5     6     7     8     9    11
  14          41    02    17    06    06    11    00    02    04    11
  15    16    25     2     3     4     5     6     7     8     9    11
  15          17    00    03    58    00    06    00    00    03    13
  16    14    26     2     3     4     5     6     7     8     9    11
  16          45    02    21    07    09    07    00    00    07    02
  17    15    10
```

TABLE 4 CONTINUED

```
17         100
18   16     27    2    3    4    5    6    7    8    9   11
18          42   00   16   27   00   00   05   00   05   05
19   14     28    2    3    4    5    6    7    8    9   11
19          47   06   19   03   03   16   00   00   03   03
20   13     29    2    3    4    5    6    7    8    9   11
20          43   00   00   10   05   05   00   00   27   10
21   14     30    2    3    4    5    6    7    8    9   11
21          58   00   10   03   00   13   00   00   03   13
22   14     31    2    3    4    5    6    7    8    9   11
22          43   03   21   03   03   08   00   03   08   08
23   14     32    2    3    4    5    6    7    8    9   11
23          44   00   28   03   00   14   00   00   03   08
24   14     33    2    3    4    5    6    7    8    9   11
24          45   03   21   07   00   14   00   00   07   03
25   14     34    2    3    4    5    6    7    8    9   11
25          50   03   18   06   03   08   00   03   06   03
26   17     35    2    3    4    5    6    7    8    9   11
26          41   00   44   03   00   03   00   00   03   06
27   13     36    2    3    4    5    6    7    8    9   11
27          61   04   15   00   04   00   00   04   08   04
@@  END OF FILE CARD
          1          1       10000
          2          2       77777
          3          3       77777
          4          4       77777
          5          5       77777
          6          6       77777
          7          7       77777
          8          8       77777
          9          9       77777
         10         10       00200
         11         11       77777
         12          1       00400
         13          1       01000
         14          1       40000
         15          1       40000
         16          1       00100
         17          1       00100
         18          1       04000
         19          1       20000
         20          1       04000
         21          1       20000
         22          1       02000
         23          1       02000
         24          1       10000
         25         18       00200
         26          1       01000
         27          1       00040
         28          1       00020
         29          1       00040
         30          1       00400
```

TABLE 4 CONTINUED

```
        31          1     00004
        32          1     00020
        33          1     00010
        34          1     00010
        35          1     00004
        36          1     00002
@@  END OF FILE CARD
01001       1.00    1  200 3.0 .25  .02
02001                     .00  .00
01002       1.00    1    1  .01  .01  .03       1.00
02002                     .00  .45
01003       1.00    1    1.01  .01                   1.00
02003                     .00  .50
01004       1.00    1    1  .01                      1.00
02004                     .00  1.10
01005        .50    1    1  .25  .25
02005                     .00  1.00
01006        .66    1    1
02006                     .00  .50
01007       .300    1    1.01         .02       1.00
02007                     .00  .25
01008       2.50    1    1.005 .005                          1.00
02008  1.00               .00  2.00
01009       1.00    1    1     .01                            1.00
02009                     .00  .25
01010      1.00 1.00 999.             1.00  1.0
02010
01011 2.50     0    1    1 .01 .02  .01       .02  .02      .02
02011                     2.50  .00
01012      1.00     130.00 1.00          1.00
02012                1.00 .00  .50
01013      .50      1    1      .20
02013      .10  .80   .90 .00  .75
01014       1.25    1    1        .25
02014      .07   .60  .80 .00  .50
01015          0    1    1
02015                 1.0      .33
01016       2.50    1    1        .25
02016      .05  .80  .85       1.00
01017       1.66    1    1        .30
02017      .02  .70  .80       1.00
01018       1.00    1  30.  .50 .50
02018
01019       1.00    1   15 1.00          1.00
02019       .00 .00  1.00
@@  END OF FILE CARD
03001      3000000000000000
03010      2000000000000000
03015      0400000000000000
03018      3000000000000000
03019      0400000000000000
@@  END OF FILE CARD
```

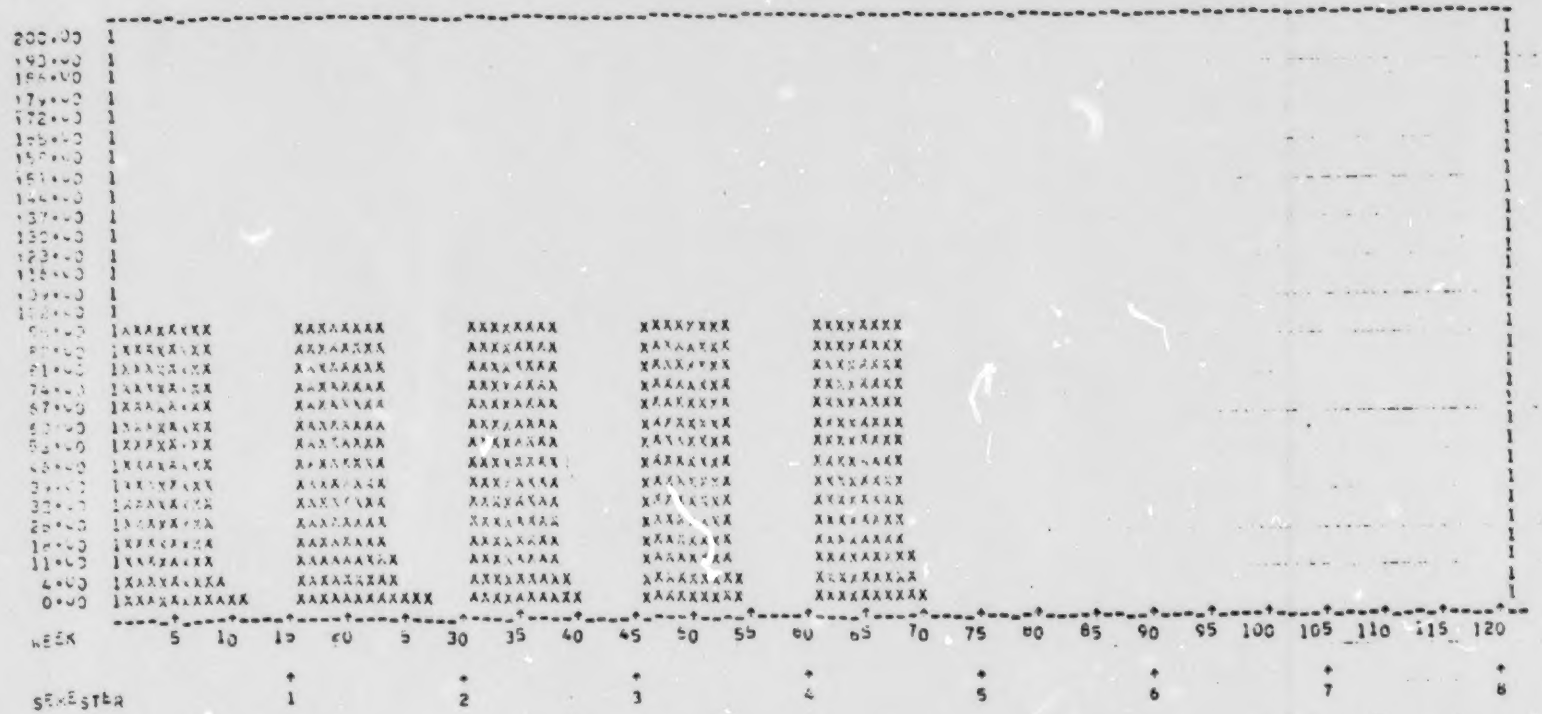
TABLE 4 CONTINUED

```
PREP HOURS PER WEEK                   1
CONTACT HOURS PER WEEK                2
LENGTH IN WEEKS                       3
STUDENTS PER IA OR PC                 4
HOURS PER WEEK AND NO. OF STAFF       5       8.0       1.0
GRADUATE ASSISTANT HOURS              6      20.0       3.0
OTHER PERSONNEL HOURS                 7      35.0       1.0
CLASSROOM HOURS                       8      40.0       1.0
LABORTORY HOURS (HRS. X STA.)         9      32.0      20.0
LIBRARY HOURS (HRS X STA)            10      96.0     100.0
MICRO TEACHING HOURS                 11      45.0       2.0
ELEMENTARY CLASSROOM HRS.            12      40.0      15.0
ELEMENTARY STUDENT HOURS             13       2.0      25.0
AUDITORIUM HOURS                     14      36.0       1.0
OFFICE HOURS                         15      10.0      99.0
OBSERVATION HOURS                    16      10.0      10.0
@@  END OF FILE CARD

      *********************************
  SOURCE OR OBJECT MAIN PROGRAM GOES HERE
  PLUS APPROPRIATE MONITOR CONTROL CARDS


  SOURCE OR OBJECT DECK FOR REPORT PROGRAM GOES HERE
  PLUS APPROPRIATE MONITOR CONTROL CARDS
      *********************************

PREP HOURS PER WEEK                   1
CONTACT HOURS PER WEEK                2
LENGTH IN WEEKS                       3
STUDENTS PER IA OR PC                 4
HOURS PER WEEK AND NO. OF STAFF       5       8.0       1.0
GRADUATE ASSISTANT HOURS              6      20.0       3.0
OTHER PERSONNEL HOURS                 7      35.0       1.0
CLASSROOM HOURS                       8      40.0       1.0
LABORTORY HOURS (HRS. X STA.)         9      32.0      20.0
LIBRARY HOURS (HRS X STA)            10      96.0     100.0
MICRO TEACHING HOURS                 11      45.0       2.0
ELEMENTARY CLASSROOM HRS.            12      40.0      15.0
ELEMENTARY STUDENT HOURS             13       2.0      25.0
AUDITORIUM HOURS                     14      36.0       1.0
OFFICE HOURS                         15      10.0      99.0
OBSERVATION HOURS                    16      10.0      10.0
@@  END OF FILE CARD


PREP HOURS PER WEEK                   1
CONTACT HOURS PER WEEK                2
LENGTH IN WEEKS                       3
```

```
                    TABLE 4  CONTINUED

STUDENTS PER IA OR PC                     4
HOURS PER WEEK AND NO. OF STAFF           5           40.00
GRADUATE ASSISTANT HOURS                  6            4.18
OTHER PERSONNEL HOURS                     7            3.30
CLASSROOM HOURS                           8             .00
LABORTORY HOURS (HRS. X STA.)             9            1.75
LIBRARY HOURS (HRS X STA)                10             .12
MICRO TEACHING HOURS                     11            6.90
ELEMENTARY CLASSROOM HRS.                12            1.00
ELEMENTARY STUDENT HOURS                 13             .00
AUDITORIUM HOURS                         14             .00
OFFICE HOURS                             15             .00
OBSERVATION HOURS                        16             .00
@@  END OF FILE CARD
```

Output 1

```
12/09/70            MODEL TEACHER EDUCATION SIMULATION PROGRAM            CASE NUMBER    2
                              NUMBER OF STUDENTS IN SYSTEM BY WEEK
```



```
          CODE      MEAN       STD.DEV.      MIN.        MAX.       OBS.

           1      54.0533      49.2448      0.0000     100.0000      75

                                NON ZERO VALUES ONLY
           1      77.9615      40.2273      1.0000     100.0000      52
```

Output 2

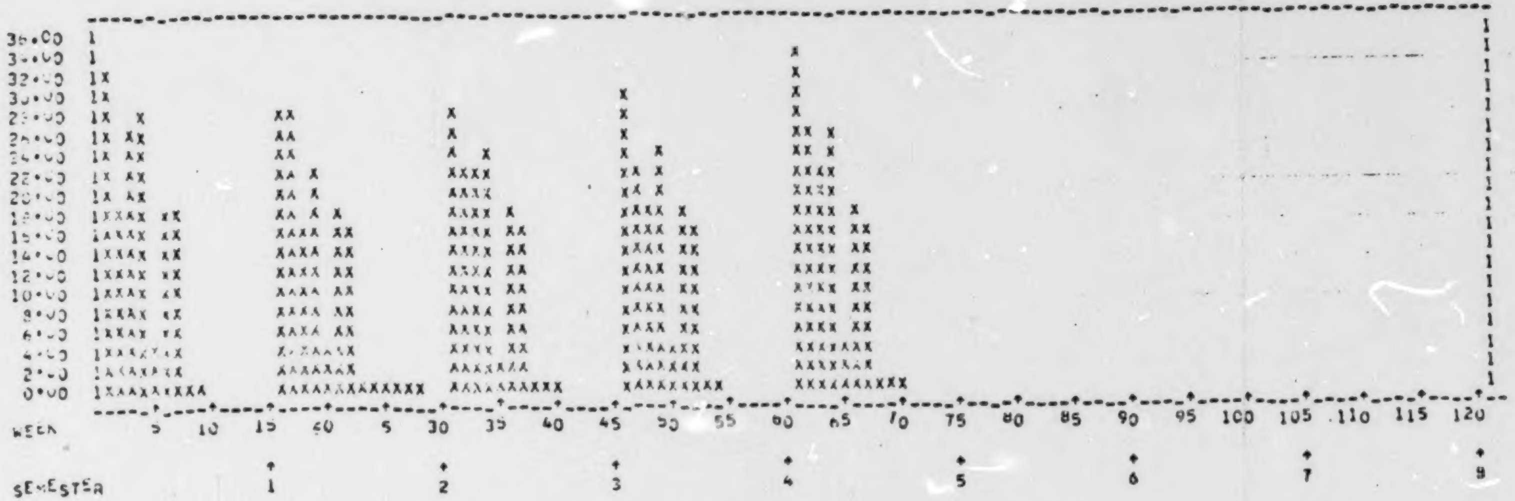
```
12/09/70                MODEL TEACHER EDUCATION SIMULATION PROGRAM                CASE NUMBER    2

                                DISTRIBUTION OF STUDENT LOADS

        HOURS            OCCURANCES            PROBABILITY

          0                  13                    0.00
          1                  41                    0.01
          2                 129                    0.03
          3                 186                    0.05
          4                 237                    0.06
          5                 265                    0.07
          6                 285                    0.07
          7                 291                    0.07
          8                 291                    0.07
          9                 285                    0.07
         10                 239                    0.06
         11                 231                    0.06
         12                 217                    0.05
         13                 189                    0.05
         14                 416                    0.10
         15                 371                    0.09
         16                 165                    0.04
         17                  91                    0.02
         18                  37                    0.01
         19                  29                    0.01
         20                  22                    0.01
         21                  14                    0.00
         22                   3                    0.00
         23                   2                    0.00
         24                   3                    0.00
         25                   0                    0.00
         26                   0                    0.00
         27                   1                    0.00

        TOTALS             4054                 1.00

          AVERAGE LOAD IS                       10.725
```

Output 3

```
12/09/70            MODEL TEACHER EDUCATION SIMULATION PROGRAM            CASE NUMBER    2

                          DISTRIBUTION OF COMPLETION TIMES

         WEEKS IN        NO OF STUDENTS        PROBABILITY OF
          SYSTEM      COMPLETING SERVICE         COMPLETION
            1                  0                    0.00
            2                  0                    0.00
            3                  0                    0.00
            4                  0                    0.00
            5                  0                    0.00
            6                  0                    0.00
            7                 10                    0.02
            8                440                    0.88
            9                 40                    0.08
           10                  7                    0.01
           11                  2                    0.00
           12                  1                    0.00

        TOTALS           500              1.00

           AVERAGE COMPLETION TIME IS              8.108
```

Output 4

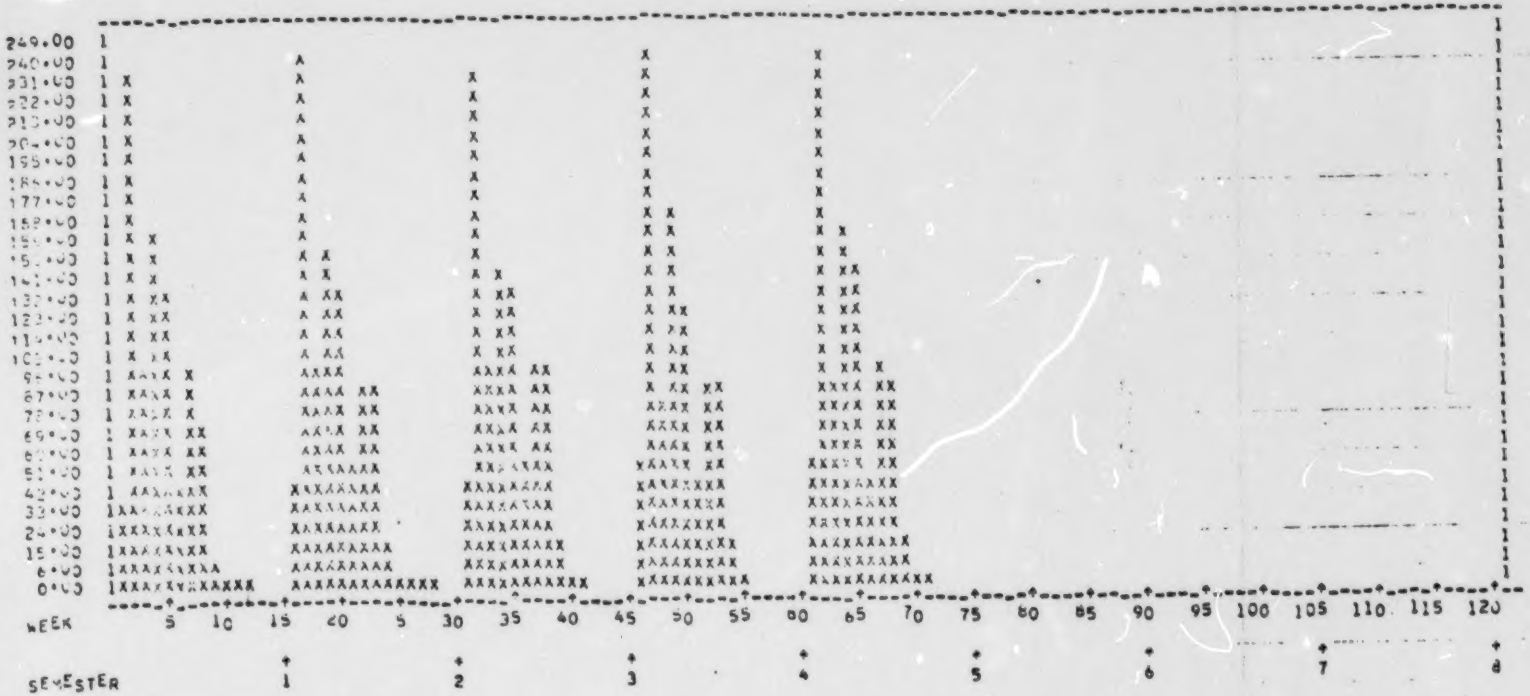
```
12/09/70              MODEL TEACHER EDUCATION SIMULATION PROGRAM              CASE NUMBER    2
                                    STAFF HOURS     UTILIZED BY WEEK
```



```
                  CODE     MEAN       STD.DEV.      MIN.        MAX.       OBS.

                   1      9.6800      11.7601      0.0000      35.6165      75

                                   NON ZERO VALUES ONLY
                   1     14.2354      11.7703      0.0036      35.6165      51

                 TOTAL RESOURCE USE        726.00
```

Output 5

```
12/09/70                    MODEL TEACHER EDUCATION SIMULATION PROGRAM                    CASE NUMBER    2
                                   GRAD ASST HRS (GTA) UTILIZED BY WEEK
```



```
                 CODE     MEAN       STD.DEV.     MIN.       MAX.      OBS.

                   1     62.0946     71.5058     0.0000    248.0826     75

                                     NON ZERO VALUES ONLY
                   1     81.7034     71.5950     0.2939    248.0826     57

                TOTAL RESOURCE USE      4657.09
```

Sample COMMON Statement For
EDSIM-4

Insert the following deck of cards
in each program or subroutine deck
of EDSIM-4 where the card appears
with COMMON punched in card columns
1 through 6.

```
                         LANGAGE ARTS COMMON
                                    STUDENT DESCRIPTORS
COMMON/1/ D(200,50 )    ,MXDR,MXDC,NSTU,NDC ,  ACTPT,ATRPT
 DIMENSION FD(200,50)  $ EQUIVALENCE  (FD,D)
 COMMON/3/STATE(200,6),MXSR,MXSC,NSAR,NSAC,INO,IPC,ISTATE
INTEGER  D,STATE,STAIA,ACTPT,ATRPT
                                    SYSTEM DESCRIPTORS
COMMON /2/PCIA(50,12),   MPBR,MPBC,NBLK ,NPBC
COMMON/5/ IAPT(100) ,MAPT, NOIA
COMMON/6/  PCPT(50) , MPC ,  NOPC
COMMON/9/ TYPE(30,25),      MIAR,MIAC,NIAR,NIAC,ITYPT
COMMON/7/  HOURS(15,20), MWKS, MHRS , NWKS , NHRS
INTEGER  PCIA,PCPT
COMMON/CTRL/ WEEK,DAY,SEM ,TIME,DT,ICASE,ETIME,STIME,NWEEK
INTEGER       WEEK,DAY,SEM ,TIME,DT,ICASE,ETIME,STIME
```

```
C
C
C
      PROGRAM SETUP
COMMON
       COMMON/10/ ITEM(48), IWORD
       COMMON/11/ ITEMP(100)
       COMMON/12/ ALOAD,ICOHORT,COHORT.STAR(20)
       COMMON /14/ ISEED(20)
C          ****************************************
C             ***************   METEP ******************
C          ****************************************
C                                       TAPE 10 INPUT DATA FOR EDSIM4
C                                       TAPE 11 USED FOR TOT. RESOURCE AVAIL.
       READ(60,1004)     ICT1        .    ICT2
      WRITE(61,1024) ICT1, ICT2
       READ (60,1006) STIME
       READ (60,1006) ETIME
       READ (60,1006) ICASE
       READ (60,1006) NWEEK
       WRITE(61,1010) STIME,ETIME       ,ICASE,NWEEK
      READ(60,1008) ALOAD
      READ(60,1006) ICOHORT
      WRITE(61,1011) ALOAD
       WRITE(61,1012) ICOHORT
       READ(60,1005) (STAR(I),I=1,16)
       WRITE(61,1009) (STAR(I),I=1,16)
       READ(60,1003)(ITEMP(I),I=1,6) ISEED( 5)
      WRITE(61,1007)(ITEMP(I),I=1,6) ISEED( 5)
       WRITE(61,1501)
C                                       SETUP RANDOM  NUMBER GENERATOR SEEDS
       DO 20 I = 1,4
   20  ISEED(I) = I*100021020
       CALL INPUT
       CALL INITIAL
C
C
       REWIND 10
       WRITE(10) ICT1,ICT2,STIME,ETIME,NSTU,ICASE,NWEEK,ALOAD,ICOHORT
       WRITE(10) ISEED.STAR
       WRITE(10)   D.STATE,         PCIA,IAPT,PCPT,TYPE,HOURS,  TRANS
        WRITE(10)                   MXDR,MXDC,NSTU,NDC ,   ACTPT.ATRPT
     1     , MXSR,MXSC,NSAR,NSAC
     2                            ,  MPBR.MPBC.NBLK .NPBC
     3                            ,MAPT, NOIA  ,MPC  ,NOPC
     4                            ,      MIAR,MIAC,NIAR,NIAC,ITYPT
     5                             , MWKS, MHRS , NWKS , NHRS
     6                            ,  MTRR,MTRC,NTRR,NTRC
        CALL DUMP
        CALL DISPLAY
       WRITE(61,1000)
 1000 FORMAT(1H1, 10X,  *  PROGRAM INITIALZED   , OUTPUT ON TAPE 10*)
 1001  FORMAT(50X, I10)
 1002  FORMAT(50X,F10.2)
 1003  FORMAT(6A8,2X,I20)
 1004 FORMAT(16I5)
 1005  FORMAT(16F5.0)
 1006 FORMAT(50X,I10)
 1007  FORMAT(1H0,5X,6A8,2X,I20)
 1008 FORMAT(50X,F10.0)
```

```
  1009  FORMAT(1H0,11X,*NUMBER OF STUDENT ARRIVALS BY SEMESTER*  /
     1 1H0,10X,16F5.0)
  1010  FORMAT(1H0,11X,*STARTING TIME OF SIMULATION IS *I10/
     11H0,10X,* ENDING TIME OF SIMULATION                *I10/
     21H0,10X,* CASE NUMBER OF SIMULATION IS             *I10/
     21H0,10X * NUMBER OF WEEKS PER SEMESTER IS          *I10)
  1011  FORMAT(1H0,11X*MAXIMUM LOAD PER STUDENT IN HOURS PER WEEK IS *
     1  F10.2)
  1012  FORMAT(1H0, 11X,*NUMBER OF STUDENTS PER COHORT IS * I10)
  1024  FORMAT(20X,*DUMP FLAG IS SET TO * I10//
     1 20X, * ARRIVAL FLAG IS SET TO * I10 )
  1501  FORMAT(1H1)
        STOP  $ END
C
C
        SUBROUTINE INPUT
COMMON
          COMMON/10/ ITEM(48), IWORD
          COMMON/11/ ITEMP(100)
C                                      READ IN PCIA FILE
C                                    AND        READ IN PC POINTER FILE
       NOP=-1
        MPC = 50
        MPBR =50    $ MPBC = 12 $ NPBC =12
        N = MPBR+1
        DO 60 I = 1,N
       READ(60,1001)NO, PCPT(I),  (PCIA(I,J),J=2,NPBC)
        IF(EOF,60) 100,45
    45 IF (NO .LE. NOP) 46,  47
    46 WRITE (61, 2001)  NOP,NO
       STOP
    47  READ(60,1001) NO2,IX, (ITEMP(J),J=1,NPBC)
       IF(EOF,60) 100,55
    55  ICNT = 0
        DO 50 J =2,NPBC
    50  IF(PCIA(I,J) .NE. 0) ICNT = ICNT + 1
        PCIA(I,1) = ICNT
        NC= ICNT + 1
       WRITE(61,1001)NO, PCPT(I),  (PCIA(I,J),J=1,NPBC)
       WRITE(61,1001) NO2,IX, (ITEMP(J);J=1,NPBC)
        IZ = 0
        DO 57 J =1,ICNT
        IZ = IZ + ITEMP(J)
    57  PCIA(I,J*1) = PCIA(I,J+1) + IZ*2**24
        IF(IZ .EQ. 100)GO TO 59
          WRITE(61,1015)IZ,(ITEMP(J),J=1,ICNT)
        STOP
    59  WRITE(61,1016) I,(PCIA(I,J),J=1,NC)
    60 CONTINUE
       WRITE (61, 2003)
        STOP
   100 NBLK = I -1
        NOPC = NBLK
C                                  READ IN IA POINTER FILE
        MAPT =  100
        N = MAPT + 1
       NOP=0
        DO 90 I = 1,N
        READ(60,1003)NO, IA, 15<53
        IF(EOF,60) 155,85
     85 IF (NOP.GE.NO)  86, 87
     86 WRITE (61, 2005)  NOP,  NO
        STOP
     87  IAPT(I) = 8H00000000
         IAPT(I)= IAPT(I) + IX*2**24 + IA*2**12
     90  WRITE(61,1007)NO, IA,IX    ,IAPT(I)
        WRITE (61, 2009)
        STOP
    155  NOIA = I - 1
C                                    READ IN RESOURCE FILE
         MIAR  = 30 $ MIAC = 25 $ NIAC = 21
         N = MIAR + 1
        NOP=0
         DO 80 I =1,N
        READ(60,1002)ITY,NO,(TYPE(I,J),J=1,15)
      1, ITY2,  NO2, (TYPE(I,J),J=16,NIAC)
        IF(EOF,60) 150,75
     75 IF (NOP.GE.NO)  76, 77
     76 WRITE (61,2007)  NOP,  NO
        STOP
     77 WRITE(61,1006)ITY,NO,  (TYPE(I,J),J=1,15)
      1 ,ITY2,  NO2, (TYPE(I,J),J=16,NIAC)
     80 CONTINUE
        WRITE (61, 2011)
        STOP
    150 NIAR = I - 1
        ITYPT =    25  $ M = 25
         N = MIAR + 1
         DO 165 I = 1,N
    156  READ(60,1012) ITY,NO,  AIX
         IF(EOF,60) 170,160
    160  WRITE(61,1013) ITY,NO,AIX
    165  TYPE(NO,ITYPT)  =AIX
         WRITE (61, 2013)
         STOP
C                                              READ IN RESOURCE CONSTRAINTS
    170  WRITE(61,1009)
          MWKS = 15  $ MHRS = 20  $ NWKS = 15
          MHRSX = MHRS +1
          DO 215 I = 1, MHRSX
          READ(60,1010)  I1,I2,I3,I4, INDEX, HRS  , ANO
          IF(EOF,60)   300,190
    190 WRITE(61,1011)  I1,I2,I3,I4, INDEX, HRS  , ANO
          IF(INDEX .GT. MHRS) GO TO 7778
          DO 200 IWK = 1,NWKS
    200   HOURS(IWK,INDEX) = HRS   * ANO
    215   CONTINUE
         WRITE (61,2015)
         STOP
    300   NHRS = I -1
          RETURN
   1001 FORMAT(16I5)
   1002 FORMAT(  I2,I3,15F5.0)
   1003  FORMAT(2I10,O10)
   1004  FORMAT(4O12)
   1005  FORMAT(1H0, 20I5)
   1006  FORMAT(1H0, 2I4,15F6.2)
   1007  FORMAT(I10,I10, 10X,O10,3X,7X,O16)
   1009  FORMAT(1H1,10X, *TYPE OF RESOURCE*,27X,*FILE POSITION*,10X,
      1           *HOURS CONTRIBUTED*,2X,          *TOTAL UNITS */
```

```
             2  80X, *PER UNIT OF*,5X,                    *OF RESOURCES*/
             3  80X, *RESOURCE *//)
      1010  FORMAT(4A8,8X,I10,2F15.0)
      1011  FORMAT(1H0, 8X, 4A8,12X,I8,12X,2F15.2)
      1012  FORMAT(I2,I3,5X,O16)
      1013  FORMAT(2I10,5X,O16)
      1014  FORMAT(1H0, *  BAD DATA BEING INPUT *)
      1015  FORMAT(*  PCIA SELECTION PBOB. SUM TO L.T. 100 *,15I5)
      1016  FORMAT(I10,2X,5O20/(12X,5O20)         )
      1501  FORMAT(1H1)
      7778  CALL ERROR(6HINPUT , INDEX, MHRS,0,0,0,0)
C
C                *      *      *     ERROR FORMAT STATEMENTS     *    *   *
C
      2001  FORMAT(1H1, * ** THE VARIABLE NO IS OUT OF ORDER.  THE PREVIOUS V
            1UE WAS,*,I5,*.    THE CURRENT VALUE IS *,I5,
            2/,10X,*THE RUN WILL BE TERMINATED  AT  SUBROUTINE INPUT.  THE ER
            3R WAS DETECTED AT LINE 45*)
      2005  FORMAT(1H1, *  THE VARIABLE NO IS OUT OF ORDER.  THE PREVIOUS V
            1UE WAS,*,I5,*.    THE CURRENT VALLE IS *,I5,
            2/,10X,*THE RUN WILL BE TERMINATED  AT  SUBROUTINE INPUT.  THE ER
            3R WAS DETECTED AT LINE 100+7*)
      2007  FORMAT(1H1, *  THE VARIABLE NO IS OUT OF ORDER.  THE PREVIOUS V
            1UE WAS,*,I5,*.    THE CURRENT VALUE IS *,I5,
            2/,10X,*THE RUN WILL BE TERMINATED  AT  SUBROUTINE INPUT.  THE ER
            3R WAS DETECTED AT LINE 155+6*)
      2003  FORMAT(1H1, //,*  THERE ARE TOO MANY P.C. S-- THEREFORE THE RUN
            1LL BE TERMINATED AT LINE  60+1 IN SUBROUTINE INPUT*)
      2009  FORMAT(1H1, //,*  THERE ARE TOO MANY I.A. S-- THEREFORE THE RUN
            1LL BE TERMINATED AT LINE  90+1 IN SUBROUTINE INPUT*)
      2011  FORMAT(1H1, //,*  THERE ARE TOO MANY RESOURCES              --THERE
            1RE THE RUN WILL BE TERMINATED AT LINE 80+1 IN SUBROUTINE INPUT*)
      2013  FORMAT(1H1, //,*  THERE ARE TOO MANY RESOURCES              --THERE
            1RE THE RUN WILL BE TERMINATED AT LINE 165+1 IN SUBROUTINE INPUT*
      2015  FORMAT(1H1, //,*  THERE ARE TOO MANY RESOURCE CONSTRAINTS--THERE
            1RE THE RUN WILL BE TERMINATED AT LINE 215+1 IN SUBROUTINE INPUT*
            RETURN        $ END
C
C
            SUBROUTINE INITIAL
COMMON
            COMMON/11/  TEMP(100)
C                                          SETUP STUDENT FILE
            ACTPT = 1    $  ATRPT = 47 $ MXDR = 200   $ MXDC = 50
            NSTU = 200   $ NDC = 50
      6400  N = ATRPT -1
      6420  DO 50 I = 1,NSTU
      6480  DO 50 J =1,N
        50  D(I,J) = 999999
C                                          SETUP STATE FILE
            MXSR = 200    $MXSC = 144 $ NSAR = NSTU   $ NSAC = NBLK
          DO 90 I =1,NSAR
            NX = MXSC/24
          DO 90 J = 1,NX
        90  STATE(I,J) = 0
            MSTC = NSTC = 53
            MSTR = NSTR = 150
C                                       CONVERT TRANSITIONAL PROBABILITIES
          MTRR = 1       $MTRC = 53
          NTRC = NBLK
      NTRR = 1
      DO 100 I = 1, NTRR
 6800 TEMP(I) = 0.0
      DO 100 J = 1,NTRC
      TRANS(I,J) = 1000/NBLK
 6860 TEMP(I) = TEMP(I) + TRANS(I,J)
  100 CONTINUE
      DO 125 I = 1,NTRR
      DO 125 J = 1,NTRC
 6940 TRANS(I,J) = TRANS(I,J)/TEMP(I)
  125 CONTINUE
 7000 N =  NTRC - 1
      DO 200 I =1,  NTRR
 7040 DO 200 J =1,N
 7060 TRANS(I,J+1) = TRANS(I,J+1) + TRANS(I,J)
  200 CONTINUE
      WRITE(61,1003)
      DO 225 I =1,15
  225 WRITE(61,2001) (HOURS(I,J),J=5,16)
C                                 DUMP TOTAL HOURS ON TAPE 11
      WRITE(11) HOURS
C                                    ADJUST RESOURCES FOR PRE SCHED ACT.
      DO 400 IA=1,NOIA
      ITYPE = IAPT(IA) .AND. 0000000077770000B
      ITYPE = ITYPE/2**12
      IF(ITYPE .EQ. 0) GO TO 400
      IF(ITYPE .LE. 0 .OR. ITYPE .GT. NIAR) GO TO 7777
C                                       CHECK FOR PRE SCHEDULED TYPES
      IX =  TYPE(ITYPE,ITYPT) .AND. 2000000000000000B
      IF(IX .EQ. 0) GO TO 400
      WEEK = 0004000000000000B
      ISCHD = IAPT(IA) .AND. 6777777700000000B
      DO 300 IW = 1, NWKS
C                                          CHECK TO SEE IF OFFERED THIS WEEK
      ICH = ISCHD .AND. WEEK
      IF(ICH .EQ. 0 ) GO TO 300
C                                          REDUCE RESOURCES
      DO 250 J = 5, NHRS
      HOURS(IW,J) = HOURS(IW,J) - TYPE(ITYPE,J)
      IF( HOURS(IW,J) .GE. 0.0)      GO TO 250
      WRITE(61,1000) IW,J,ITYPE
      STOP
  250 CONTINUE
  300 WEEK = WEEK/2
  400 CONTINUE
      WRITE(61,1004)
      DO 500 I =1,15
  500 WRITE(61,2001) (HOURS(I,J),J=5,16)
 7100 RETURN
 1000 FORMAT(1H0,10X, * RESOURCE NOT AVAILABLE TO CONDUCT PRESCHED.
     1 ACTIVITY * 4I10)
 1001 FORMAT(1H0,10X, * TYPE CODE TO LARGE OR SMALL * I10)
 1003 FORMAT(1H1,30X, *  TOTAL RESOURSES AVAILABLE */)
 1004 FORMAT(1H1,15X,*ADJUSTED RESOURCES AFTER REQUIREMENTS HAVE BEEN*
     1* MET FOR  FOR PRESCHEDULED IA */)
 2001 FORMAT(15F8.2)
 7777 WRITE(61,1001) ITYPE
      STOP
 7120 END
C
```

```
C
      SUBROUTINE DUMP
C     COMMON
      EQUIVALENCE (IWORD,WORD)
      COMMON/10/ ITEM(48), IWORD
      COMMON/11/ ITEMP(100)
C                                       PCIA TABLE OUT
      LINE = 0
      DO 100 I = 1,NBLK
      IF(LINE -(LINE/25) * 25  .EQ. 0) 58,59
   58 WRITE(61,1008)
     WRITE (61,1000)
     WRITE(61,1068) (K,K=1,9)
   59  LINE = LINE + 1
      KNT = PCIA(I,1) + 1
      DO 60 J = 2,KNT
      ITEM(J-1) = PCIA(I,J) .AND. 0000000077777777B
   60 ITEMP(J-1) = PCIA(I,J)/2**24
      KN = KNT - 1
      WRITE(61,1001) I,PCPT(I), PCIA(I,1),(ITEM(J),J=1,KN )
      WRITE(61,1010) ( ITEMP(J), J=1,KN )
  100 CONTINUE
C                                        PRINT IAPT FILE
      LINE = 0
      DO 200 IN = 1,NOIA
      IF(LINE -(LINE/50) * 50  .EQ. 0) 119,128
  119 WRITE(61,1069)
     WRITE (61,1071)
     WRITE (61,1002) (JK,JK=1,15)
  128 LINE = LINE + 1
     I = IN
      NO = (IAPT(I) .AND. 0000000000007777B)
      IA = (IAPT(I) .AND. 0000000077777777B )
      IA = IA/2**12
     IWORD = IAPT(I)
      CALL BITPK
      WRITE(61,1003) IN,IA,  (ITEM(I),I=10,24)
  200 CONTINUE
C                                  PRINT RESOURCE REQUIREMENTS FILE
      LINE = 0
      DO 400  I = 1, NIAR
      IF(LINE -(LINE/30) * 30  .EQ. 0) 388,399
  388 WRITE (61,1006)
     WRITE (61,1076)
     WRITE (61,1077)  (KK,KK=1,19)
  399 LINE = LINE + 1
      DO 390 J = 1,NIAC
  390 IF(TYPE(I,J) .EQ. 0.0) TYPE(I,J) = 0.0
      WRITE(61,1007)I,(TYPE(I,J) , J=1,NIAC)
  400 CONTINUE
      WRITE(61,1089)
      DO 500 I = 1,NIAR
       WORD = TYPE(I,ITYPT)
      CALL BITPK
      WRITE(61,1009)I,( ITEM(J),J=1,8)
  500 CONTINUE
      WRITE(61,1501)
      DO 250 I =1,15
  250 WRITE(61,2001) (HOURS(I,J),J=5,16)
      WRITE(61,1501)
      CALL MPRT(TRANS,MTRR,MTRC,NTRR,NTRC,1)
 1000 FORMAT (1H0,5X,* PC NUMBER *23X, *INSTRUCTIONAL ALTERNATIVES*,//
     1 51X,      *(NUMBER) *   )
 1001 FORMAT(1H , 5X,3I8,2X,15I5,/(1H0,31X,15I5))
 1002 FORMAT(39X,15I3,/)
 1003 FORMAT(1H ,3X,2I10,15X,24I3)
 1004 FORMAT(1H ,13X,I20,10X,I20)
 1005 FORMAT(1H1,34X,*P C   P O I N T E R   F I L E*//
     1  32X, *P C *,23X,* PC TYPE*,//)
 1006 FORMAT(1H1,35X, *R E S O U R C E   R E Q U I R E M E N T S   F I L
     1E  *,//)
 1007 FORMAT(1H0,       I5 ,   2X,  21F6.2  )
 1008 FORMAT(1H1,39X,*P. C. - I. A.    F I L E *    )
 1009 FORMAT(1H0, I10,20I5)
 1010 FORMAT(1H , 31X, 15I5)
 1068 FORMAT (19X,*(TYPE)  *,      *(NO.)*,20I5)
 1069 FORMAT(1H1,22X,*SCHEDULE OF INSTRUCTIONAL ALTERNATIVES*,//)
 1071 FORMAT (11X,* IA NO.   IA TYPE      *,25X,*(WEEK) *)
 1076 FORMAT (3X,*TYPE* ,51X,*(FILE POSITION)*)
 1077 FORMAT (3X,*CODE*,3X,20(I4,2X),/)
 1088 FORMAT (1H1)
 1089 FORMAT(1H1,20X,*  POSITION 1 SIGN BIT   */
     1                21X,*  POSITION 2 PRESCHEDULED IA = 1 */
     1                21X,*  POSITION 3 SERV. TIME FIXED = 1 */
     1                21X,*  POSITION 4 EXPRESIVE OBJECTIVE = 1*/)
 1501 FORMAT(1H1)
 2001 FORMAT(15F8.2//)
15750 RETURN           $END
C
C
      SUBROUTINE BITPK
      COMMON/10/ ITEM(48), IWORD
      ITEM(1) = 0
      MASK = 2000000000000000B
      DO 100 I = 2,48
      IX = MASK .AND. IWORD
      ITEM(I) = 0
      IF(IX .NE. 0) ITEM(I) = 1
  100 MASK = MASK/2
      RETURN   $ END
C
C
      SUBROUTINE ERROR(I,A,B,C,D,E,F)
      COMMON/CTRL/ WEEK,DAY,SEM ,TIME,DT,ICASE,ETIME,STIME,NWEEK
      INTEGER      WEEK,DAY,SEM ,TIME,DT,ICASE,ETIME,STIME
      WRITE(61,1001), TIME,I,A,B,C,D,E,F
      WRITE(61,1002), TIME,I,A,B,C,D,E,F
 1002 FORMAT (* ERROR  *I10,A6,6I10)
 1001 FORMAT ( *  ERROR *,I10,A6,6F15.2)
      RETURN $ END
C
C
```

```
C
      PROGRAM EDSIM4
COMMON
        COMMON/10/ STAT(30)
        COMMON/12/ ALOAD.ICOHORT.COHORT.STAR(20)
        COMMON/13/ ITP(50)
        COMMON /14/ ISEED(20)
       COMMON/15/ OUT(200)
        COMMON/16/ TEMP(15,20), SUMA(30,5)
        COMMON/17/ RUSE(12,15)
                                    TAPE 10 INPUT FROM INITIALIZE
C                                   TAPE 11 TEMPORY OUTPUT FOR HOURS
C                                          ICT1 IS TAPE DUMP FLAG
C                                          ICT2 IS SYSTEM DUMP FLAG
C
C        ********************            START     ***************************
        REWIND 11
        READ(11) TEMP
       REWIND 10
        READ(10) ICT1.ICT2,STIME.ETIME.NSTU.ICASE,NWEEK.ALOAD,ICOHORT
        WRITE(61.1C03)ICT1.ICT2,STIME.ETIME,NSTU.ICASE,NWEEK.ALOAD,
      1 ICOHORT
        READ(10) ISEED.STAR
       READ (10)   D,STATE              ,PCIA,IAPT,PCPT.TYPE.HOURS, TRANS
        READ (10)                   MXDR,MXDC,NSTU.NDC ,   ACTPT.ATRPT
      1     . MXSR.MXSC.NSAR.NSAC
      2                             .  MPBR.MPBC.NBLK ,NPBC
      3                            ,MAPT, NOIA   ,MPC ,NOPC
      4                             .       MIAR,MIAC,NIAR,NIAC,ITYPT
      5                               . MWKS, MHRS . NWKS . NHRS
      6                             .  MTRR,MTRC,NTRR,NTRC
       WRITE(   10    )   STIME, ETIME,  DT, NSTU,ICASE,NWEEK
 C *******************INITIALIZE SYSTEM *************************************
         SEM   = STIME     $NSAR = NSTU       $ NSTR = NSTU
         REWIND 11
        ISEM = SEM
         WRITE(11) HOURS
         RATE = STAR(ISEM)
         NSTU = 0
         COHORT = ICOHORT
         TIME = 1
         DT = TIME
         CALL ARRIVE(RATE,ICT2)
         WEEK = 00040000000000006
        WRITE(61,1000) TIME.DT.ICASE.ETIME.STIME.SEM.NSTU
         DO 5 I =1,15
         DO 5 J =1,5
         SUMA(I,J) = 0.0
         SUMA(I,4) = 9999999.9
       5 SUMA(I,5) =-9999999.9
         GO TO 95
  C ***************************** SETUP TIME COUNTER ********************
      10 CALL TIMER(ISTOP)
          IF(ISTOP .NE. 0) GO TO 11
          IF(TIME   -(TIME/NWEEK)*NWEEK .NE. 1) GO TO 11
        ISEM = SEM
         RATE = STAR(ISEM)
         CALL ARRIVE(RATE,ICT2)
  C   11 WRITE(61,1000) TIME.DT.ICASE.ETIME.STIME.SEM.NSTU
      11  CONTINUE
  C      ************************         SCAN  FOR COMPLETIONS **************
```

```
   15 ISTU = 0
   50 ISTU = ISTU + 1
C                             ZERO OUT STUDENT LOAD
       FD(ISTU,ATRPT+2) =   0.0
      IF(ISTU .GT. NSTU)   GO TO 95
   45 CALL SCAN (ISTU,IFLG)
C                                    IF IFLG EQ 0 MEANS STUDENT HAS NOT COMLET
      IF(IFLG .EQ. 0) GO TO 50
      CALL PASS(ISTU,IFLG,IPASS)
C                              UPDATE STUDENT RECORDS
      CALL UPDATE(ISTU,IFLG,IPASS)
       GO TO 45
C  *****************     SELECT ACTIVITIES  ***********************
   95 IF(ISTOP .NE. 0)   GO TO 500
      ISTU = 0
C                             SELECT PROCESS BLOCK
  100 IC = 1
  110 ISTU = ISTU + 1
       IF(ISTU .GT. NSTU) GO TO 401
       IXX = 999
      CALL CLOD (ISTU,IFLG2)
C                                   STUDENT HAS FULL LOAD THIS TIME PERIOD
       IF(IFLG2 .EQ. 7777) GO TO 100
  120 CALL SELBLK(ISTU,IBLK)
      IF(IBLK .EQ. 88888) GO TO 100
C                                        88888 NO BLKS COMPLETED
C                                               OR NO ROOM IN D FILE OR STUDENT
C                                               STU. CMPLT. OR ENGAGED IN ALL
C                             77777 ALL BLKS COMPLETED
      IF(IBLK .NE. 77777) GO TO 200
C                             REMOVE STUDENT FROM THE SYSTEM
       CALL REMOVE(ISTU,ISTOP)
       ISTU = ISTU - 1
       GO TO 100
C                                  SELECT ACTIVITY
C                                      PRETEST STUDENT
  200  CALL PTEST(ISTU,IBLK,IPASS)
       IF(IPASS .EQ. 0) GO TO 210
C                                    STUDENT PASSES PRETEST
       CALL UPDAT2(ISTU,IBLK,IPASS)
       STAT(12) = STAT(12) + 1
       GO TO 120
  210 CALL SELA3T(ISTU,IBLK,IACT,IFLG1,ITYPE,ISWK)
C                        888888 MEANS CAP OF PRE SCHED. IA REACHED OR
C                        RESOURCES NOT AVAILABLE TO COND. IA
       IF(ITYPE .NE. 888888 ) GO TO 300
  220  IC = IC + 1
       IF(IC .LE. 3  ) GO TO 120
       STAT(9) = STAT(9) + 1
        GO TO 100
  300  CALL SCHED   (ISTU,IACT,IBLK,ITYPE,ISWK)
       IF(IBLK .EQ. 99999) GO TO 220
C                       7777 CAP OF STU. REACHED (HRS./WEEK)
       IF(IBLK .EQ. 7777) GO TO 100
       GO TO 120
C********************** TAPE DUMP AND END OF LOOP ******************
  401  CONTINUE
C                                         CHECK DUMP FLAGS
       IF(ICT1 .EQ. 0)WRITE (10) DT,TIME,SEM,NSTU,D,STAT,OUT,HOURS
       IF(ICT2 .NE. 0) CALL DUMP(0,0,1,1,1,0)
      GO TO 10
C                *******************       END OF PROGRAM RUN ***************
  500 WRITE(61,1001) TIME   ,ISTOP,SEM
       CALL DATACLT
      ENDFILE 10
      REWIND 10
      CALL DUMP(0,0,1,1,1,0)
      STOP
C        SSSS******************** ERROR DUMP ****************************
 7778 CALL ERROR(6HMAIN    ,ISTU,NSTU,IC,NWLK,0,0,2)
 1001  FORMAT( 10X, * END OF SIMULATION RUN AT * 5I10)
 1000 FORMAT(10X,*  SIMULATION TIME IS *  10I8)
 1003  FORMAT( 10I10)
 1004 FORMAT(16I5)
 1005  FORMAT(10X, *WEEK MASK IS  *  O16)
 1006  FORMAT(2X,30F4.0)
      END
C
C
      SUBROUTINE TIMER(ISTOP)
COMMON
       COMMON/10/ STAT(30)
      STOP = 0
       TIME = TIME + 1
       DT = TIME -((TIME-1)/NWEEK)*NWEEK
       WEEK = WEEK/2
       DO 210 I = 1,30
  210  STAT(I) = 0.0
       IF(TIME   -(TIME/NWEEK)*NWEEK .NE. 1) GO TO 7777
       SEM = SEM + 1
       IF(SEM .GT. ETIME) ISTOP = 1
  100  REWIND 11
       READ(11) HOURS
       WEEK = 00040000000000000B
 7777   RETURN  $ END
C
C
       SUBROUTINE SCAN(ISTU,IFLG)
COMMON
       INTEGER EDWK,STWK
       COMMON/10/ STAT(30)
       COMMON/13/ TP(50)
 8320  IF(ISTU .GT. NSTU .OR. ISTU .LE. 0) GO TO 7778
C                                 SCAN FOR COMPLETION OF ACTIVITY
       IFLG = 0
 8360  M=ACTPT+2
 8380  N=ATRPT-1
C      ******************  START MAIN LOOP **************************
 8400  DO 20 J=M,N,3
       IX = D(ISTU,J)
       EDWK =(IX .AND.  00000000777777778)
       IX = IX/2**24
       STWK =(IX .AND.  00000000777777778)
       IF(STWK .GT. TIME) GO TO 20
       IF(EDWK .GT. TIME) GO TO 20
C                        IF END WK. LESS THAN TIME SET FLG
 8440  IFLG=J
       GO TO 7777
   20  CONTINUE
       IFLG = 0
 7777  RETURN
 9778 CALL ERROR(4HSCAN,ISTU,NSTU,ITYPE,NIAR,IACT, J,1)
      CALL ERROR(4HSCA2 ,IWK,NW,0,0,0,0,1)         $ RETURN
       END
 C
 C
       SUBROUTINE PASS (ISTU,IPT,IPASS)
COMMON
       COMMON/10/ STAT(30)
       COMMON /14/ ISEED(20)
 C                                 IPASS  = 0  FAILS PC
 C                                IPASS = 1    PASSES PC
       INT = 0
       IBLK =D(ISTU,IPT-2)
       IF(IBLK .LE. 0 .OR. IBLK .GT. NOPC ) GO TO 7778
       ITYPE = PCPT(IBLK)
       IF  (ITYPE .LE. 0 .OR. ITYPE .GT. NIAR) GO TO 7778
       PPASS = TYPE(ITYPE,19)
   80  IPASS=1
 C                                 CHECK FOR EXPRESSIVE OBJECTIVE
 C                                 NO RESOURCES REC. EXP. IN IA
       IX =  TYPE(ITYPE,ITYPT) .AND. 04000000000C0000B
       IF(IX .NE. 0) GO TO 7777
       CALL RANFSET(ISEED(1))
 9100  X = RANF(-1)
 9120  IF(X.GE.PPASS) IPASS=0
       CALL RANFGET(ISEED(1))
 C                                                  UPDATE STUDENT TIME REQ.
       IX =  TYPE(ITYPE,ITYPT) .AND. 1C000000C0000000B
       EX = TYPE(ITYPE,2) + TYPE(ITYPE,1)
       X = EX
       IF(IX .NE. 0) GO TO 50
       STDX = SQRT(TYPE(ITYPE,20)**2 + TYPE(ITYPE,21)**2)
       CALL SRVTM(EX,STDX,X )
   50  FD(ISTU,ATRPT+2) =FD(ISTU, ATRPT +2) +     X
 C                                REDUCE RESOURCES TO CONDUCT P C
       ALOD = X/EX
       CALL USE(ITYPE,DT,DT,ALOD ,1,6)
       IACT = 2*INT+1+IPASS
 7777  RETURN
 7778  CALL ERROR(6HPASS    ,IBLK,ITYPE,ISTU,IPT,ANO,INT, 1)  $ RETURN
 C ***********************  ENTRY POINT FOR PRETEST ******************
       ENTRY PTEST
       INT = 1
       IBLK = IPT
       IF(IBLK .LE. 0 .OR. IBLK .GT. NOPC ) GO TO 7778
       ITYPE = PCPT(IBLK)
       IF  (ITYPE .LE. 0 .OR. ITYPE .GT. NIAR) GO TO 7778
 C                                    TEST FOR PRETEST
       IPASS = 0
      PPASS = TYPE(ITYPE,17)
       IF(PPASS .LE. 0.0) RETURN
       Y = RANF(-1)
       IF(X.GE.PPASS)   RETURN
 C                STUDENT WILL PRETEST SETUP FOR PRETEST
       PPASS =  TYPE(ITYPE,18)
       GO TO 80
 9160  END
 C
 C
```

```
      SUBROUTINE UPDATE(ISTU,IPT,IPASS)
COMMON
      COMMON/12/ ALOAD,ICOHORT,COHORT,STAR(20)
      IFLG = 0
C                   UPDATE STATE IF STUDENT PASSES PROCESS BLOCK
      IACT = D(ISTU,IPT-1)
      IF(IACT .LE. 0 .OR. IACT .GT. NOIA ) GO TO 7778
 9540 IBLK=D(ISTU,IPT-2)
   10 IF(IBLK .LE. 0 .OR. IBLK .GT. NOPC ) GO TO 7778
      ISTATE = 0
      INO = ISTU $ IPC = IBLK  $  CALL GET
      IF(ISTATE .EQ. 1) GO TO 7778
      IF(IPASS.EQ.1) GO TO 50
      ISTATE = 0 $ INO = ISTU $  IPC = IBLK $  CALL PUT
      GO TO 100
   50 ISTATE = 1 $ INO = ISTU $  IPC = IBLK $  CALL PUT
        IF(IFLG .EQ. 11) GO TO 7777
C                                       RESET TIME
  100 D(ISTU,IPT) = 999999
C                                              REDUCE NUMBER OF STUDENTS IN IA
      ITA = IAPT(IACT) .AND. 00000000000077778
      IF(ITA-1  .LT. 0) GO TO 7778
      IAPT(IACT) = IAPT(IACT) - ICOHORT
 7777     RETURN
 7778 CALL ERROR(6HUPDATE,IBLK,IPT,IPASS,IACT,ITYPE,ISTU,       1)
      CALL ERROR(6HUPDAT1,ISTATE,0,0,0,0,0,0)
      CALL ERROR(6HUPDAT2,ITA,IACT,ISTU,IPT,IFLG,ISTATE,2)
C  ********  ENTRY POINT FOR UPDATE PRETEST STUDENTS ****************
      ENTRY UPDAT2
      IBLK = IPT
      IFLG = 11
      GO TO 10
 9900 END
C
C
      SUBROUTINE CLOD(ISTU,IFLG2)
COMMON
      INTEGER EDWK,STWK
      COMMON/10/ STAT(30)
      COMMON/13/ TP(50)
      COMMON/14/ ISEED(20)
 8320 IF(ISTU .GT. NSTU .OR. ISTU .LE. 0) GO TO 7778
C                                   SCAN FOR COMPLETION OF ACTIVITY
      IFLG2 = 0
      DO 5 I = 1,50
    5 TP(I) = 0.0
      MX=ACTPT+2
      NX=ATRPT-1
C     *******************  START MAIN LOOP **************************
      DO 200  J=MX,NX,3
      IX = D(ISTU,J)
      EDWK =(IX .AND.  00000000777777778)
      IX = IX/2**24
      STWK =(IX .AND.  00000000777777778)
      IACT = D(ISTU,J-1)
C                                            CHECK FOR END OF ACTIVE PART R
      IF(IACT .EQ. 999999) GO TO 200
      IF(IACT .LE. 0 .OR. IACT .GT. NOIA) GO TO 7778
      IF(D(ISTU,J) .EQ. 999999) GO TO 200
      ITYPE = (IAPT(IACT)  .AND. 00000000777700008)
```

```
          ITYPE = ITYPE/2**12
          IF(ITYPE .LE. 0 .OR. ITYPE .GT. NIAR ) GO TO 7778
C                                LOAD IS CONTACT PLUS PREP TIME
          EX = TYPE(ITYPE,2) + TYPE(ITYPE.1)
          IX =  TYPE(ITYPE.ITYPT) .AND. 1000000000000000B
          N = EDWK-1
          M = STWK
          IF(STWK .LT. TIME) M = TIME
          DO 45 IWK = M,N
          NW = IWK -((IWK-1)/NWEEK) * NWEEK
          IF(NW  .GT. NWEEK) GO TO 7778
          X = EX
          IF(IX .NE. 0) GO TO 45
          STDX = SQRT(TYPE(ITYPE.20)**2 + TYPE(ITYPE,21)**2)
          CALL SRVTM(EX.STDX.X )
       45 TP(NW ) = TP(NW ) + X
      200 CONTINUE
C     ****************   END OF DO LOOP ***************************
          XLD = TP(DT)
          FD(ISTU.ATRPT + 2) =  FD(ISTU.ATRPT +2) + XLD
          IF(XLD .LT. FD(ISTU.ATRPT + 1) ) GO TO 7777
          IFLG2 = 7777
          STAT(7) = STAT(7) + 1
     7777 RETURN
     7778 CALL ERROR(4HCLOD.ISTU.NSTU.ITYPE.NIAR.IACT. J.1)
          CALL ERROR(4HCLO2 .IWK.NW.ATRPT ,ACTPT .0.0.1)       $ RETURN
     8540 END
C
C
          SUBROUTINE      SELBLK(ISTU, IBLK)
COMMON
          COMMON/10/ STAT(30)
          COMMON /14/ ISEED(20)
          IF(ISTU  .GT. NSTU ) GO TO 7778
C                                CHECH FOR END TIME OF 999999
          N=ATRPT-1
          M=ACTPT+2
C                                             SCAN FOR FIRST EMPTY CELL
          DO 20 J=M,N,3
          IF(D(ISTU,J) .EQ.999999) GO TO 10
       20  CONTINUE
C                               STUDENT HAS NOT COMPLETED ANY BLKS
C                               DURING LAST TIME PERIOD
C                              OR NO ROOM IN FILE FOR NEW PC
          STAT(8) = STAT(8) + 1
          IF(FD(ISTU.ATRPT+2) .LT. FD(ISTU.ATRPT+1))STAT(10)=STAT(10)+1
          IBLK = 88888
          GO TO 7777
       10 CALL RANFSET(ISEED(2))
           X = RANF(-1)
          CALL RANFGET(ISEED(2))
C                                I - BLOCK JUST COMPLETED
          I = 1
C                                SELECT NEXT BLOCK
    10700 DO 100 J =1.NTRC
    10720 IF(X .LE. TRANS(I.J)) GO TO 200
      100 CONTINUE
         GO TO 7778
      200 IBLK = J
          JX = J
```

```
C                                 CHECK TO SEE IF BLOCK COMPLETED
          IF(IBLK .GT. NBLK) GO TO 7778
          INO = ISTU $ IPC = IBLK  $  CALL GET
          IF( ISTATE .EQ. 0 ) GO TO 7777
C                                     CHECH TO SEE IF ALL BLOCKS COMPLETED
C                                             OR IN PROGRESS
          IBLK = 77777
C                                       SELECT NEXT PC IN SEQUENCE
          N = NSAC - 1
         DO 300 J = 1,N
          JX = JX + 1
          JX = JX -((JX-1)/NBLK)*NBLK
          INO = ISTU  $ IPC = JX$ CALL GET
          ISTAT = ISTATE
          IF(ISTAT .NE. 1)  IBLK = 88888
          IF(ISTAT .EQ. 1 .OR. ISTAT .EQ. 2) GO TO 300
          IBLK = JX
          IF(IBLK .GT. NBLK) GO TO 7778
          GO TO 7777
    300 CONTINUE
          STAT(11) = STAT(11) + 1
C                              ALL BLOCKS COMPLETED IF IBLK = 77777
   7777 RETURN
   7778 CALL ERROR(6HSELBLK,ISTU,IBLK,NBLK,X,I,J,1)    $ RETURN
  10960  END
C
C
          SUBROUTINE SELACT(ISTU,IBLK,IACT,IFLG1,ITYPE,ISWK)
COMMON
          COMMON/10/ STAT(30)
          COMMON/12/ ALOAD,ICOHORT,COHORT,STAR(20)
          COMMON/13/ TP(50)
          COMMON /14/ ISEED(20)
C                                  DOES NOT CHECK TO SEE IF STUDENT HAS
C                              ENGAGED IN ACTIVITY BEFORE
          ITRY = 0
          IFLG1 = 000
          AMLD = FD(ISTU,ATRPT +1)
          CRLD = FD(ISTU,ATRPT+2)
C                                  RANDOMLY SELECT IA
          CALL RANFSET(ISEED(3))
          IX = RANF(-1)* 100. + 1.0
          CALL RANFGET(ISEED(3))
          ICNT = PCIA(IBLK,1)
          N = ICNT + 1
          DO 20 IP = 2,N
          IPROB = PCIA(IBLK,IP)/2**24
          IF(IX .LE. IPROB) GO TO 15
     20   CONTINUE
          GO TO 7778
     15   IF(IP.LE. 1 .OR. IP .GT. NPBC) GO TO 7778
          IACT = P31A(IBLK,IP) .AND. 00000000777777778
          IF(IACT .LE. 0 .OR. IACT .GT. NOIA ) GO TO 7778
          ITA = IAPT(IACT)
          IX = (    ITA      .AND. 77777777000000000B)
          IXMK = IX
          ISCD = (IX .AND. WEEK)
          ITYPE = (  ITA          .AND. 00000000777700000B)
          ITYPE = ITYPE/2**12
          IF(ITYPE .LE. 0 .OR. ITYPE  .GT.  NIAR) GO TO 7778
```

```
C                                    IS THIS PRE SCHEDULED IA
         IX =  TYPE(ITYPE.ITYPT) .AND. 2000000000000000B
         HR = TYPE(ITYPE.1) + TYPE(ITYPE.2)
         IF(IX .EQ. 0) GO TO 200
C                                    IS IA OFFERED THIS TIME PERIOD
         IF(ISCD .EQ. 0)  GO TO 100
C                                                      CHECK CAP OF STUDENT
         IF(CRLD + HR .GT. AMLD)  GO TO 100
C                                         CHECK CAPACITY OF IA
         ISWK = DT
    32   ICAP = TYPE(ITYPE.4)
         INO = ITA .AND. 0000000000077777B
         IF(INO+ICOHORT  .LE. ICAP) GO TO 7775
         STAT(15) = STAT(15) + 1
C                                    CAP. OF IA EXCEDED
C                                          ATTEMPT TO SCHEDULE ALTERNATIVE I.
    45   ITRY = ITRY + 1
         IF(ITRY .GE. ICNT) GO TO 50
         IP = IP + 1
         IF(IP .GT. ICNT+1) IP = 2
         GO TO 15
C                                         RECORD  DISAPPOINTMENT
    50   STAT(1) = STAT(1) + 1
         ITYPE = 888888
         GO TO 7777
C                                    SCAN FUTURE TIME PERIODS FOR IA OFFERREIN
   100    IWK = WEEK
         NNX = DT + 1
         NMX = NNX + ISEED(5)-1
         IF(NMX .GT. NWEEK) NMX = NWEEK
         DO 60 J = NNX.NMX
         IWK = IWK/2
         ISCD = IXMK .AND. IWK
         IF(ISCD .EQ. 0) GO TO 60
         ISWK =  J
         IF(ISWK .LE. 0 .OR. ISWK .GT. NWKS) GO TO 7778
C                                    CHECK LOAD DURING WEEK J
         IF(TP(J) + HR .GE. AMLD) GO TO 60
         GO TO 32
C                                    LOAD ON STU. GT MAX FOR TIME PERIOD
    60   CONTINUE
         STAT(13) = STAT(13) + 1
         GO TO 45
C*********************************        CHECK CAPACITY OF RESOURCES ***
C                                         INDIVIDUAL STUDY
C                                    IS IA OFFERED THIS TIME PERIOD
   200   IF(ISCD .EQ. 0) GO TO 45
         IF(CRLD + HR .GT. AMLD)  GO TO 45
         ISWK = DT
         IF(ISWK .LE. 0 .OR. ISWK .GT. NWKS) GO TO 7778
C                                            CHECK AVAIL. OF RESOURCES
        ISTOP = ISWK + TYPE(ITYPE,3)  -1
         CALL USE(ITYPE, ISWK.ISTOP,1.0.0.3)
  7775   CONTINUE
C                                    RESOURCES NOT AVAILABLE IF ITYPE EQ 88888
  7777   RETURN
  7778  CALL ERROR(6H$ELACT   .ISTU.IELK.IACT,IP,1.X.1)
        CALL ERROR(6H$ELAC2,ITYPE,NIAR,ISWK,IPROB,IX,ISCD,1)  $ RETURN
 11540   END
C
```

```
C
      SUBROUTINF SCHED(ISTU,IACT,IBLK,ITYPE,ISWK)
COMMON
      COMMON/10/ STAT(20)
      COMMON/12/ ALOAD,ICOHORT,COHORT,STAR(20)
C                          STORE BLOCK AND ACTIVITY NUMBERS
12681 IF(ISTU .GT. NSTU)  GO TO 7778
12821 IF(IACT .GT. NOIA) GO TO 7778
     IF(IBLK .GT. NOPC) GO TO 7778
      INO = ISTU $ IPC = IBLK  $  CALL GET
      IF(ISTATE .NE. 0) GO TO 7778
      N=ATRPT-1
      M=ACTPT+2
      DO 20 J=M,N,3
      IF(D(ISTU,J) .EQ.999999) GO TO 10
   20 CONTINUE
12760 GO TO 7778
   10 D(ISTU,J-2)=IBLK
      D(ISTU,J-1)=IACT
     IWK = TYPE(ITYPE,3)
      IF(IWK .LE. 0 .OR. ISWK .LE. 0) GO TO   7778
C                                        LOAD STARTING WEEK
      IBGN = (SEM-1)*NWEEK+ISWK
      IF(IBGN .LT. TIME) GO TO 7778
      D(ISTU,J) = IBGN*2**24
C                                        LOAD ENDING WEEK
      D(ISTU,J) = D(ISTU,J) + IWK+ IBGN
      IF(IBGN .GT. TIME) GO TO 55
C                                          CHECK SERVICE TIME
      HR = TYPE(ITYPE,2) + TYPE(ITYPE,1)
      SVHR = HR
      IX =  TYPE(ITYPE,ITYPT) .AND. 1000000000000000B
      IF(IX .NE. 0) GO TO 50
C                                        RANDOM SERVICE TIME
      EX = HR
      STDX = SQRT(TYPE(ITYPE,20)**2 + TYPE(ITYPE,21)**2)
      CALL SRVTM(EX,STDX,HR)
C                                                 UPDATE STUDENT LOAD
C                                 NO OF HRS PER WEEK
   50 FD(ISTU,ATRPT+2) =FD(ISTU, ATRPT +2) +   HR
C                                          SET STATE TO (2)
C*********************************** AJUST RESOURCE LOADS *********
   55 ITA = IAPT(IACT) .AND. 0000000000077778
      ISTATE = 2 $ INO = ISTU $  IPC = IBLK $  CALL PUT
C                                          CHECK FOR PRE SCED IA
      IX =  TYPE(ITYPE,ITYPT) .AND. 2000000000000000B
      IF( IX .NE. 0) GO TO 120
      IF(ISWK .LE. C .OR. ISWK .GT. NWKS) GO TO 7778
C                                          UPDATE RESOURCE FILE
     ALOD = HR/SVHR
      ISTOP = ISWK + IWK - 1
      CALL USE(ITYPE, ISWK, ISTOP, ALOD,1,16)
C                                          UPDATE NO. STUD IN IA
  120 ITA = ITA + ICOHORT
      IF(ITA .GT. 0000000000077778) GO TO 7778
      IAPT(IACT) = IAPT(IACT) + ICOHORT
C                                CHECK LOAD ON STUDENT
      IF(FD(ISTU,ATRPT+2) .LT. FD(ISTU,ATRPT +1 ))  GO TO 7777
      IBLK = 7777
       STAT(14) = STAT(14) + 1.
```

```
 7777  RETURN
 7778 CALL ERROR(5HSCHED  ,ISTU,IACT,IBLK,NSTU,ITA,IBGN,1)
      CALL ERROR(5HSCHE2, ISTATE,0,0,0,0,0,1)
       CALL ERROR(5HSCHD2  , IWK,ISWK, ITYPE, J,M,N,  1) $ RETURN
13120  END
C
C
       SUBROUTINE USE( ITYPE,ISTART,ISTOP,ALOD,IFLG,ICALL)
COMMON
       COMMON/10/ STAT(30)
       COMMON/12/ ALOAD,ICOHORT,COHORT,STAR(20)
      ANO = TYPE(ITYPE,4)
       IF(ANO .LT. 1.0) GO TO 7778
       COHORT = ICOHORT
       FAC =ALOD*COHORT/ANO
       IRS = ITYPE
       DO 300 I     = ISTART, ISTOP
       IPER = I - ((I- 1)/NWKS)*NWKS
       DO 300 J =5,16
      AX = HOURS(IPER,J) - TYPE(ITYPE,J)*FAC
       IF(AX .GE. 0.0) GO TO 190
       AX = 0.0
       IRS = 888258
       STAT(ICALL) = STAT(ICALL) + 1
      STAT(J+12) = STAT(J+12) + 1
  190  IF(IFLG .EQ. 0) GO TO 300
       HOURS(IPER,J) = AX
  300  CONTINUE
       ITYPE = IRS
 7777  RETURN
 7778  CALL ERROR(6HUSE    ,ANO,ITYPE,IFLG,ICALL,ALOD,ISTART,1) $ RETUR
       END
C
C
       SUBROUTINE ARRIVE(RATE,IFLG1)
COMMON
       COMMON/10/ STAT(30)
       COMMON/12/ ALOAD,ICOHORT,COHORT,STAR(20)
      COMMON/15/ OUT(200)
       NOIN = RATE
   60  IF(NSTU+NOIN .GT. MXSR ) NOIN = MXSR -NSTU
       M = NSTU + 1
       NSTU = NSTU + NOIN
       NSAR = NSTR = NSTU
       N = ATRPT -1
       DO 105 I= M,NSTU
       DO 100 J=1,N
  100  D(I,J) = 999999
       FD(I,ATRPT+1) = ALOAD
       FD(I,ATRPT+2) = 0.0
        D(I,ATRPT+3)   =TIME
       STAT(4) = STAT(4) + ICOHORT
  105  D(I,ATRPT) =ICOHORT
       RETURN $ END
C
C
       SUBROUTINE REMOVE(ISTU,ISTOP)
COMMON
       COMMON/10/ STAT(30)
       COMMON/12/ ALOAD,ICOHORT,COHORT,STAR(20)
      COMMON/15/ OUT(200)
C                                 TABULATE TOTAL TIME IN SYSTEM
      IT = D(ISTU,ATRPT+3)
      ID = TIME -IT
       IF(ID .LE. 0 .OR. ID .GT.190 ) GO TO 7778
      TIM = ID
      CALL COLCT(TIM,27)
      CALL COLCT(TIM,1)
      IF(IT .LE. 15) GO TO 10
      OUT(ID) = OUT(ID) + ICOHORT
      OUT(200) = OUT(200) + ICOHORT
   10 STAT(5) = STAT(5) + ICOHORT
C                                  ELIMINATE STUDENT ENTRY IN D FILE
      DO 20 J = 1,NDC
   20 D(ISTU,J) = D(NSTU,J)
      DO 50 J = 1,NSAC
      INO = NSTU  $  IPC = J  $  CALL GET
      INO = ISTU  $  IPC = J  $  CALL PUT
      ISTATE = 0
      INO = NSTU  $  IPC = J  $  CALL PUT
   50 CONTINUE
      NSTU = NSTU - 1
      NSAR =NSTR=NSTU
      RETURN
 7778  CALL ERROR(6HREMOVE ,ISTU,IT,TIME,ID,0,0,2)
      RETURN  $ END
C
C
      SUBROUTINE DUMP(I1,I2,I3,I4,I5,I6)
COMMON
      COMMON/13/ ITP(50)
      INTEGER EDWK,STWK
     WRITE(61,1000) TIME   ,SEM
15460 IF(I1 .EQ. 1) GO TO 10
15470 PRINT 8000
      DO 100    I = 1,NSTU
      K = 1
      DO 90 J =1,21,3
      ITP(K) = D(I,J)
      ITP(K+1) = D(I,J+1)
      IX = D(I    ,J+2)
      EDWK =(IX .AND.  00000000777777775)
      IX = IX/2**24
      STWK =(IX .AND.  00000000777777778)
      ITP(K+2) = STWK
      ITP(K+3) = EDWK
      K = K + 4
   90 CONTINUE
      IAX =FD(I,ATRPT+1)
      IAY =FD(I,ATRPT+2)
      WRITE(61,1010)I,(ITP(J),J=1,28)   ,IAX,IAY
  100 CONTINUE
   10 IF(I2 .EQ. 1) GO TO 20
15560 PRINT 8002
      DO 150 I = 1,NSTU
      IF(I-(I/30)*30  .EQ. 1) WRITE(61,1001)I2,(J,J=1,50)
      INO = I
      DO 145 J =1,50
      IPC = J
      CALL GET
  145  ITP(J) = ISTATE
       WRITE(61,1001) I,(ITP(J) ,J=1,30)
  150  CONTINUE
   20  IF(13 .EQ. 1) GO TO 30
15510  PRINT 8001
       DO 175 I = 1,NBLK
  175  WRITE(61,1013) I , (PCIA(I,J),J=1,NPBC)
       DO 176 I =1,MIAR
  176  WRITE(61,1008)(TYPE(I,J),J=1,MIAC)
   30  IF(14 .EQ. 1) GO TO 40
      WRITE(61,1002)( I, PCPT(I), I=1,NOPC)
       DO 500 I =1, NOIA
       IX = IAPT(I) /2**24
       NO = (IAPT(I) .AND. 0000000000007777B)
       IA = (IAPT(I) .AND. 00000000777777778 )
       IA = IA/2**12
      WRITE(61,1004) I,IX,IA,NO    ,IAPT(I)
  500  CONTINUE
   40  IF(15 .EQ. 1) GO TO 50
   50  IF(16 .EQ. 1) GO TO 60
       WRITE(61,1006)
       WRITE(61,1007)    DT, (HOURS(DT,J),J=5,16)
   60  CONTINUE
 1000  FORMAT(1H1, 20X, * SYSTEM DUMP AT  *5I10  //)
 1001  FORMAT(1H0,10X,I10,5X,30I2)
 1002  FORMAT(20X,10I10///)
 1004  FORMAT(10X, I5, 2X,O16, I10,I10,3X,O16)
 1005  FORMAT(1X, 10E12.2)
 1006  FORMAT(20X, * RESOURSE STATUS * )
 1007  FORMAT(1H0,I10,12F10.2)
 1008  FORMAT(1H0,20F6.2)
 1010  FORMAT(1H0,I5,32I4)
 1013  FORMAT(1H ,I10,2X,30I20/(10X,30I20))
 8000  FORMAT (* STUDENT FILE *)
 8001  FORMAT(*  PCIA   IAPT   PCPT  *  )
 8002  FORMAT(* STATE FILE*)
15750  RETURN          $END
C
C
       SUBROUTINE SRVTM(EX,STDX,X)
       COMMON /14/ ISEED(20)
       AMIN = EX -3.0*STDX
C       MIN SER. TIME ASSUMED TO BE 5 MIN OR .0833 HRS
       IF(AMIN .LT. .0833333) AMIN = .0833333
       AMAX = EX + 3.0*STDX
       SUM = 0.0
       CALL RANFSET(ISEED(4))
       DO 5 I = 1,12
       R = RANF(-1)
    5  SUM =SUM + R
       CALL RANFGET(ISEED(4))
       X = STDX*(SUM-6.0)+ EX
       IF(X.GT. AMAX) X = AMAX
       IF(X .LT. AMIN) X = AMIN
       RETURN $ END
C
C
          IDENT      XDATA
    3     BLOCK
          COMMON     DATA(200,6),L,M,XXX,ZZZ,I,J,IDATA
          ENTRY     GET
 GET      UBJP      **
          SIU       GRETN,1
          SIL       GRETN,2
          RTJ       GADDR
 +        LBYT,A0,E2,CL   DATA,1,2              LOAD BYTE FROM ARRAY
          STA       IDATA
 GRETN    ENI       **,1
          ENI       **,2
          SLJ       GET
          ENTRY     PUT
 PUT      UBJP      **
          SIU       PRETN,1
          SIL       PRETN,2
          RTJ       GADDR
 +        LDA       IDATA
          SBYT,A0,E2 DATA,1,2               STORE BYTE IN ARRAY
 PRETN    ENI       **,1
          ENI       **,2
          SLJ       PUT
 GADDR    SLJ       **
          LDA       K
          INA       -1
          MUI       M
          ADD       J
          INA       -1
          MUI       L
          ADD       I
          INA       -1
          ENQ       0
          DVI       =D24                 CELLS PER 48 BITS
          RXT       A,B1
          RXT       Q,A                   BITS PER CELL
          MUI       =D2
          ENQ       46
          ROP,-     Q,A,A
          RXT A,B2
          SLJ       GADDR
 N        DEC       1
 K        DEC       1
          END
 C
 C
       SUBROUTINE ERROR(I,A,B,C,D,E,F,ISTOP )
       COMMON/CTRL/ WEEK,DAY,SEM ,TIME,DT,ICASE,ETIME,STIME,NWEEK
       INTEGER      WEEK,DAY,SEM ,TIME,DT,ICASE,ETIME,STIME
        ICNT = ICNT + 1
       WRITE(61,1002), TIME,I,A,B,C,D,E,F
  1002 FORMAT (* ERROR  *I10,A6,6I10)
       WRITE(61,1001), TIME,I,A,B,C,D,E,F
  1001 FORMAT ( *  ERROR *,I10,A6,6F15.2)
        IF(ICNT .LT. 100) RETURN
        WRITE(61,1000)
  1000  FORMAT(10X,/////  *  SIMULATION TERMINATED DUE TO ERROR   *   )
       CALL DUMP(0,0,0,0,0,0)
        STOP
        END
 C
 C
```

```
     C
           PROGRAM REPORT
     COMMON
             COMMON/10/ STAT(20)
             COMMON/11/ ALCAD,ICOHORT,COHORT
             COMMON/12/  LOAD(150),  AMIN(150),  AMAX(150),STLD(150)
             COMMON /13/ FINIS(150). ITEMP(10) . ASTU(150).LDZ00(150)
             COMMON/14/ TEMP(15,20),IPAGE
             COMMON /15/ ISEED(20),STAR(20)
            COMMON/16/ OUT(200)
             COMMON/17/ COST(20), TCOST
             COMMON/18/ SELTP(100), SELIA(30,12)
                                 *****************************************
      C                          ******     METEP      ******************
      C                          *****************************************
      C
             REWIND 9
            REWIND 10
                         TAPE 9 USED TO STORE HOURS
      C                  TAPE 10 INPUT TAPE TO REPORT GENERATOR
      C
            CALL TITLE
                                 READ IN TAPE HEADER INFORMATION
      C
             READ(10) ICT1,ICT2,STIME,ETIME,NSTU,ICASE,NWEEK,ALOAD,ICOHORT
             READ(10) ISEED   ,STAR
            READ (10)    D,STATE          ,PCIA,IAPT,PCPT,TYPE,HOURS, TRANS
             READ (10)                  MXDR,MXDC,NSTU,NDC ,  ACTPT,ATRPT
          1    , MXSR,MXSC,NSAR,NSAC
          2                       ,  MPBR,MPBC,NBLK ,NPBC
          3                      ,MAPT, NOIA  ,MPC ,NOPC
          4                       ,      MIAR,MIAC,NIAR,NIAC,ITYPT
          5                          , MWKS, MHRS , NWKS , NHRS
          6                       ,  MTRR,MTRC,NTRR,NTRC
             READ(  10   )  STIME, ETIME,  DT, NSTU,ICASE,NWEEK
            CALL INPUT
             TCOST = 0.0
             COHORT = ICOHORT
             DO 8 I =1,MPBR
             DO 8 J =1,MPBC
          8  SELIA(I,J) = 0.0
         10  READ (10) DT,TIME,SEM,NSTU,D,STAT,OUT,HOURS
             IF(EOF,10)  500,15
         15  CALL TAB
             WRITE(9) HOURS, TIME, SEM   ,DT
            GO TO 10
                     ******************     END OF PROGRAM RUN **************
       C
        500 WRITE(61,1001) TIME
            CALL REPORT1
             ENDFILE 9
             REWIND 9
             CALL REPORT2
            REWIND 10
             WRITE(61,1002) TCOST
            STOP
                 $$$$******************** ERROR DUMP **************************
       C
       7778 CALL ERROR(6HMAIN   ,ISTU,NSTU,IC,NBLK,0,0)
             STOP
       1000 FORMAT(10X,*  SIMULATION TIME IS *  1018)
       1001  FORMAT(1H1,10X,*  END OF SIMULATION RUN AT * 5110)
       1002  FORMAT(1H1,* TOTAL COST OF ALL RESOURCES * F18.2)
            END
       C
```

```
C
      SUBROUTINE INPUT
COMMON
        COMMON/14/ TEMP(15,20),IPAGE
        COMMON/17/ COST(20), TCOST
        WRITE(61,1009)
        MHRSX = MHRS +1
        DO 215 I = 1, MHRSX
        READ(60,1010)  I1,I2,I3,I4, INDEX, HRS   , ANO
        IF(EOF,60)   300,190
  190 WRITE(61,1011)  I1,I2,I3,I4, INDEX, HRS   , ANO
        DO 200 IWK = 1,NWKS
  200    TEMP(IWK,INDEX) = HRS   * ANO
  215   CONTINUE
  300   WRITE(61,1008)
        DO 400 I =1,MHRSX
        READ(60,1010)I1,I2,I3,I4, INDEX, COST(INDEX)
        IF(EOF,60) 410,400
  400 WRITE(61,1011)I1,I2,I3,I4, INDEX, COST(INDEX)
  410   CONTINUE
 1008   FORMAT(1H1, 55X,* COST PER UNIT OF RESOURCE * )
 1009   FORMAT(1H1,10X, *TYPE OF RESOURCE*,27X,*FILE POSITION*,10X,
    1          *HOURS CONTRIBUTED*,2X,                *TOTAL UNITS */
    2    80X, *PER UNIT OF*,5X,                       *OF RESOURCES*/
    3    80X, *RESOURCE *//)
 1010   FORMAT(4A8,3X,I10,2F15.0)
 1011   FORMAT(1H0, 8X, 4A8,12X,I8,12X,2F15.2)
 1501   FORMAT(1H1)
      RETURN    $    END
C
C
        SUBROUTINE   TAB
COMMON
        COMMON/10/ STAT(30)
        COMMON/11/ ALOAD,ICOHORT,COHORT
        COMMON/12/  LOAD(150),  AMIN(150),  AMAX(150),STLD(150)
        COMMON /13/ FINIS(150), ITEMP(10) , ASTU(150),LDZOO(150)
        COMMON/14/ TEMP(15,20),IPAGE
        COMMON/18/ SELTP(100), SELIA(50,12)
        REAL LOAD ,LDZOO
        INTEGER EDWK,STWK
        IT = TIME
        IF(IT .LE.150) GO TO 25
        CALL ERROR  (6H TAB    ,IT,TIME,STIME,DT,0,0)
        GO TO 7777
   25 CNT = 0.0
        AMIN(IT)  =999999999.9
        AMAX(IT) = -99999999.9
        TOT = 0.0
        IF(NSTU .EQ. 0)  AMIN(IT) = 0.0
        ANO = NSTU
        DO 50 I=1,NSTU
        ALOAD =FD(1,ATRPT+2)
C                                       TAB NUMBER OF STUDENTS WITH ZER
        IF(ALOAD .LE. .95) CNT = CNT + 1.0
        TOT = TOT + ALOAD
        IF(ALOAD .LT. AMIN(IT)) AMIN(IT) = ALOAD
        IF(ALOAD .GT. AMAX(IT) ) AMAX(IT) = ALOAD
C                                       TAB LOADS BY NUMBER OF HOURS
        ILD = ALOAD    + 1.0
      IF(ILD .GT. 0 .AND. ILD .LE. 150) GO TO 40
      WRITE(61,1003) I,NSTU, ALOAD
      GO TO 50
   40 STLD(ILD) = STLD(ILD) + 1.0
   50 CONTINUE
      LDZOO(TIME) = CNT/COHORT
      IX=IT
                                            STORE AVERAGE LOAD
C
      LOAD(IX) = TOT/ ANO                   STORE NUMBER OF COMPLETIONS
C
     FINIS(IT) = STAT(5)
                                  TAB. NO. OF STUDENTS IN SYSTEM
C
       ASTU(IT) = NSTU * COHORT
                                ACCUMLATE INSTRUCTIONAL ALTERNATIVE STAT.
C
      MX=ACTPT+2
      NX=ATRPT-1
      DO 300 ISTU =1,NSTU
      DO 300  J=MX,NX,3
      IX = D(ISTU,J)
      EDWK =(IX .AND.  00000000777777778)
      IX = IX/2**24
      STWK =(IX .AND.  00000000777777778)
      IF(STWK  .NE. TIME ) GO TO 300
      IACT = D(ISTU,J-1)
      IBLK = D(ISTU,J-2)
      IF(IACT .EQ. 999999) GO TO 300
      IF(IACT .LE. 0 .OR. IACT .GT. NOIA) GO TO 7778
      ITYPE = (IAPT(IACT)  .AND. 00000000777700008)
      ITYPE = ITYPE/2**12
      IF(ITYPE .LE. 0 .OR. ITYPE .GT. NIAR ) GO TO 7778
                                          ACCUMLATE IA BY TYPE
C
      SELTP(ITYPE) = SELTP(ITYPE) + 1.0
                                   ACCUMUATE BY PC TYPE
C
      ITYPE = PCPT(IBLK)
      IF(ITYPE .LE. 0 .OR. ITYPE .GT.NOPC) GO TO 7778
      SELTP(ITYPE) = SELTP(ITYPE) + 1.0
                                            STORE IA BY PC
C
      ICNT = PCIA(IBLK,1)
      DO 275  IA = 1,ICNT
      IAT =  PCIA(IBLK,IA+1) .AND.  00000000777777778
      IF(IAT               .EQ. IACT) GO TO 280
  275 CONTINUE
      WRITE(61,1004) IAT,IACT,IBLK,ISTU
      GO TO 300
  280 SELIA(IBLK,IA  ) = SELIA(IBLK,IA  ) + 1.0
  300 CONTINUE
                                  PLOT DISAPPOINTMENTS COUNTERS
C
      IF(LINE -(LINE/45)*45 .NE. 0) GO TO 200
      IF(IT .EQ. 1) IPAGE = 1
      CALL PAGE(IPAGE)
      WRITE(61,1000)
      WRITE(61,1002) IT,IT,(J,J=1,30)
  200 WRITE(61,1001) TIME,SEM,(STAT(I),I=1,30)
      LINE = LINE + 1
      RETURN
 1000 FORMAT(1H0,55X, *DATA COUNTERS                    *//)
 1001 FORMAT(1H  ,2I4,*  I * , 30F4.0)
 1002 FORMAT(1H  , 2I6,30I4 //)
 1003 FORMAT(*  STUDENT HAS LOAD G.T. 150 HRS *  2I10,F10.2)
 1004 FORMAT(  *      ERROR 5L/4TAB IA NOT FOUND IN PCIA FILE * 4I10)
 7778 CALL ERROR(4HTAB .ISTU.NSTU.ITYPE.NIAR.IACT, J.1)
      CALL ERROR(4HTAS  .IWK.NW.ATRPT .ACTPT .0.0.1)          $ RETURN
 7777  RETURN $ END
C
C
        SUBROUTINE REPORT1
COMMON
        COMMON/10/ STAT(30)
        COMMON/11/ ALOAD.ICCHORT.COHORT
        COMMON/12/  LOAD(150).  AMIN(150).  AMAX(150).STLD(150)
        COMMON /13/ FINIS(150). ITEMP(10) . ASTU(150).LDZ00(150)
        COMMON/14/ TEMP(15.20).IPAGE
       COMMON/16/ OUT(200)
        COMMON/18/ SELTP(100), SELIA(50,12)
        REAL LOAD ,LDZ00
        TLD = 30.00
        DELTA = 1.00
        CALL PAGE( IPAGE )
        WRITE(61,1001)
        CALL GRAPH(150,LOAD,1H*,1H ,TLD . 0.0  . DELTA )
        CALL PAGE( IPAGE )
        WRITE(61,1002)
        CALL GRAPH(150.AMIN.1H*,1H .TLD . 0.0  . DELTA      )
        CALL PAGE( IPAGE )
        WRITE(61.1003)
        CALL GRAPH(150.AMAX.1H*.1H .TLD . 0.0  . DELTA )
        CALL PAGE( IPAGE )
        WRITE(61,1004)
        ANSTU = MXDR*ICCHORT
        DELTA = MXDR * ICCHORT / 30  + 1
        CALL GRAPH(150,FINIS,1H*,1H . ANSTU.0.0.   DELTA)
        CALL PAGE( IPAGE )
        WRITE(61,1005)
        CALL GRAPH(150. ASTU.1HX.1H .ANSTU.0..DELTA)
        CALL PAGE(IPAGE)
        WRITE(61.1011)
        CALL GRAPH(150.LDZ00.1HX.1H .ANSTU. 0.0.DELTA)
        CALL PAGE(IPAGE)
        WRITE(61,1012)
        CALL GRAPH(150.STLD.1HX.1H . ANSTU.0.0.DELTA)
        SUM = 0.0
        TOT = 0.0
       CALL PAGE(IPAGE)
        WRITE(61,1006)
        WRITE(61.1008)
       DO 200 I=1,190
        PROB = OUT(I)/OUT(200)
        WRITE(61,1009) I. OUT(I) . PROB
        SUM = SUM + PROB
        IF (SUM .GE. 1.00) GO TO 220
  200   CONTINUE
  220   WRITE(61.1010) OUT(200) . SUM
        SUM = 0.0        $ NX = 1
        DO 223 J =1.NX
        AJ = J
  223   SUM = SUM +AJ*OUT(J)
        AMEAN = SUM/OUT(200)
        WRITE(61.1018) AMEAN
        CALL PAGE(IPAGE)
        WRITE(61.1012)
      WRITE(61,1013)
      TOT = 0.0
      DO 250 I =1,150
  250 TOT = TOT + STLD(I)
      SUM = 0.0
      DO 275 I =1,150
      PROB : STLD(I)/TOT
      SUM = SUM + PROB
      IXX = I - 1
      WRITE(61,1009) IXX , STLD(I), PROB
      IF( SUM .GE. 1.00) GO TO 280
  275 CONTINUE
  280 WRITE(61,1010) TOT, SUM
      SUM = 0.0        $ NX = 1
      DO 283 J =1,NX
      AJ = J
  283 SUM = SUM + AJ*STLD(J)
      AMEAN = SUM /TOT
      WRITE(61,1018) AMEAN
C                    WRITE SELECTION I.A. REPORT
      WRITE(61,1014)
      DO 300 I = 1,NIAR
  300 WRITE(61,1015) (I,SELTP(I))
      WRITE(61,1016)
      DO 340 I = 1,NBLK
  340 WRITE(61,1017) (I,(SELIA(I,J),J=1,NPBC) )
 1000 FORMAT(1H0,20F6.2)
 1001 FORMAT(1H0, 55X, *AVERAGE STUDENT LOAD BY WEEK  * )
 1002 FORMAT(1H0, 55X, *MINIMUM STUDENT LOAD BY WEEK *)
 1003 FORMAT(1H0, 55X, *MAXIMUM STUDENT LOAD BY WEEK *)
 1004 FORMAT(1H0, 55X, *NUMBER OF COMPLETIONS BY WEEK *)
 1005 FORMAT(1H0,55X,*NUMBER OF STUDENTS IN SYSTEM BY WEEK *)
 1006 FORMAT(1H0,55X,*DENSITY FUNCTION OF TIME TO COMPLETION* )
 1007 FORMAT(20X,F20.5)
 1008 FORMAT(1H0,30X,*WEEKS IN*7X*NO OF STUDENTS*6X*PROBABILITY OF*/
     1 32X *SYSTEM* 6X *COMPLETING SERVICE* 6X *COMPLETION*)
 1009 FORMAT(1H , 30X,I5,15X,F4.0,15X, F4.2)
 1010 FORMAT(1H0,22X, *TOTALS* 19X, F7.0, 9X, F10.2)
 1011 FORMAT(1H0,35X, *NUMBER OF STUDENTS WITH ZERO LOADS *  )
 1012 FORMAT(1H0,55X, * DISTRIBUTION OF STUDENT LOADS * )
 1013 FORMAT(1H0,30X,*HOURS*10X*OCCURANCES* 10X,*PROBABILITY*//)
 1014 FORMAT(1H1,//40X,* INSTRUCTIONAL ALTERNATIVE SELECTION DISTRIBUT
     1ON * // 30X,*  TYPE  * 10X, * NUMBER SELECTING *  )
 1015 FORMAT(1H ,31X,I10,10X,F12.2)
 1016 FORMAT(1H1,//,40X,*INSTRUCTIONAL ALTERNATIVE SELECTION BY P.C.*
     1//,20X,* P. C. *, 10X, *DISTRIBUTION OF IA *//)
 1017 FORMAT(1H , 20X, I7, 10X, 15F6.0)
 1018 FORMAT(1H0,22X, *MEAN IS *  10X,  F10.3)
      RETURN  $ END
C
C
      SUBROUTINE REPORT2
COMMON
      COMMON/12/  LOAD(150),  AMIN(150),  AMAX(150),STLD(150)
      COMMON /13/ FINIS(150), ITEMP(10) , ASTU(150),LDZOO(150)
      COMMON/14/ TEMP(15,20),IPAGE
      COMMON/17/ COST(20), TCOST
      REAL LOAD ,LDZOO
C                                   READ IN INITIAL HOURS ARRAY
C                              TABULATE RESOURCE LOADS
```

```
          DO 200 JC = 5,NHRS
           IF(JC  .EQ.17) GO TO 200
           IF(JC  .EQ.18) GO TO 200
           IF(JC  .EQ.19) GO TO 200
           IF(JC  .EQ. 1) GO TO 200
           IF(JC  .EQ. 2) GO TO 200
           IF(JC  .EQ. 3) GO TO 200
           IF(JC  .EQ. 4)  GO TO 200
           CALL PAGE( IPAGE )
           IF(JC  .EQ. 5) WRITE(61,1005)
           IF(JC  .EQ. 6) WRITE(61,1006)
           IF(JC  .EQ. 7) WRITE(61,1007)
           IF(JC  .EQ. 8) WRITE(61,1008)
           IF(JC  .EQ. 9) WRITE(61,1009)
           IF(JC  .EQ.10) WRITE(61,1010)
           IF(JC  .EQ.11) WRITE(61,1011)
           IF(JC  .EQ.12) WRITE(61,1012)
           IF(JC  .EQ.13)  WRITE(61,1001)
           IF(JC  .EQ.14)   WRITE(61,1002)
           IF(JC  .EQ.15)   WRITE(61,1003)
           IF(JC  .EQ.16) WRITE(61,1004)
           USE = 0.0
           REWIND 9
           AMAX = 0.0
          CNT =0.0
           DO 75 I = 1,150
           ASTU(I) = 0.0
     75    FINIS(I) = 0.0
            DO 100 IT = 1,1000
           READ(  9) HOURS, TIME, SEM      ,DT
           IF(EOF, 9) 110, 70
     70    ITX = TIME
           IF(ITX .LT. 0  .OR. ITX .GT. 150) GO TO 7778
           FINIS(TIME) = TEMP(DT  ,JC) - HOURS(DT  ,JC)
           USE = USE + FINIS(TIME)
           ASTU(TIME) = FINIS(TIME)/TEMP(DT,JC)
          PER = FINIS(TIME)/TEMP(DT,JC)
          IP = PER *100.
           IF(IP .EQ. 0) GO TO 78
          IF(IP.GT.100 .OR.IP .LE. 0) GO TO 77
           GO TO 78
     77    WRITE(61,1025) IP,PER, FINIS(TIME), TEMP(DT,JC), HOURS(DT,JC)
     78    HRS = FINIS(TIME)
           IF(AMAX .LT. HRS ) AMAX = HRS
    100    CONTINUE
C                                                 END OF LOOP
           WRITE(61,1000) JC
           STOP
    110    IF(AMAX .LT. 30. ) AMAX = 29.0
           IT = AMAX
           AMAX = IT + 1
           ID = (IT-1)/30  + 1
           DELTA = ID
           CALL GRAPH(150,FINIS,1HX,1H ,AMAX, 0., DELTA)
           RCOST = USE* COST(JC)
           WRITE(61,1018) USE, COST(JC), RCOST
           TCOST = TCOST + RCOST
C                                           TABULATE PERCENT USE OF RESOURCE
          CALL PAGE(IPAGE)
           WRITE(61,1017)
          WRITE(61,1020)
          DV = TIME
          AXX = .00
          SUM = 0.0
          DO 300 INX = 1,10
          TOT = 0.0
          DO 180 I = 1,TIME
          BXX = AXX + .10
          IF(BXX .GT. .96) BXX = 1.0009
  180     IF(ASTU(I) .GE. AXX .AND. ASTU(I) .LT. BXX) TOT = TOT + 1.0
          PROB = TOT/DV
          SUM = SUM + PROB
          WRITE(61,1019) AXX,BXX,TOT, PROB
          AXX = AXX + .10
  300     CONTINUE
          WRITE(61,1026) DV,SUM
          WRITE(61,1021) (ASTU(I),I =1,TIME)
                                   END OF DO LOOP FOR COMPLETE SUBROUTINE
C
 200     CONTINUE
 7777     RETURN
 1000     FORMAT(1H0,   * NO EOF ON TAPE 9 *  I10)
 1001 FORMAT (1H0,55X, *ELEM. PUPIL HRS.    UTILIZED BY WEEK      * )
 1002 FORMAT (1H0,55X, *AUDITORIUM HOURS    UTILIZED BY WEEK      * )
 1003 FORMAT (1H0,55X, *OFFFICE HOURS       UTILIZED BY WEEK      * )
 1004 FORMAT (1H0,55X, *OBSERVATION HRS.    UTILIZED BY WEEK      * )
 1005 FORMAT (1H0,55X, *      STAFF HOURS   UTILIZED BY WEEK      * )
 1006 FORMAT (1H0,55X, *GRAD ASST HRS (GTA) UTILIZED BY WEEK      * )
 1007 FORMAT (1H0,55X, * LAB TECH HRS       UTILIZED BY WEEK      * )
 1008 FORMAT (1H0,55X, *CLASSROOM HOURS     UTILIZED BY WEEK      * )
 1009 FORMAT (1H0,55X, *          LAB HOURS UTILIZED BY WEEK      * )
 1010 FORMAT (1H0,55X, *  LIBRARY HOURS     UTILIZED BY WEEK      * )
 1011 FORMAT (1H0,55X, *MOTION PICTURE HRS  UTILIZED BY WEEK      * )
 1012 FORMAT (1H0,55X, *ELEM. CLASSRM HRS   UTILIZED BY WEEK      * )
 1015 FORMAT (1H0,55X, *MICRO TEACH. HOURS  UTILIZED BY WEEK      * )
 1017 FORMAT(1H0,55X,* DISTRIBUTION OF RESOURCE LOADINGS* )
 1018  FORMAT(1H0,25X, *TOTAL RESOURCE USE *,F12.2,2X, *COST PER UNIT*
     1,F10.2,2X, *TOTAL COST * F12.2)
 1019  FORMAT(1H , 30X, F6.2, F8.2, 10X,F10.0,10X,F7.3)
 1020  FORMAT(1H0, 30X,*LOADING OF RESOURCE* 25X, *PROBABILITY*/
     1          33X, *(PERCENT)*)
 1021  FORMAT(10X,15F7.2)
 1025  FORMAT(* BAD DATA IN REPORT2* I10,5F15.2)
 1026  FORMAT(1H0,31X, *TOTALS * ,12X,F14.0,10X,F7.3///////)
 7778 CALL ERROR (6HREPT2 ,ITX,SEM,NWEEK,0,0,0)
      RETURN  $   END
C
C
       SUBROUTINE SDIV(AA,NN)
       DIMENSION AA(NN)
       IFLG = 1
    5  SUM4 = 9999999.9
       SUM1 = 0.0
       SUM2 = 0.0
       SUM3 = 0.0
       SUM5  = -999999.9
   10  DO 100 I = 1,NN
       X = AA(I)
       IF(IFLG .EQ. 1) GO TO 50
       IF(ABS(X) .LT.  .000001) GO TO 100
   50  SUM1 = SUM1 + X
```

```
            SUM2 = SUM2 + X*X
            SUM3 = SUM3 +1.0
            SUM4 = AMIN1(SUM4,X)
            SUM5 = AMAX1(SUM5,X)
    100     CONTINUE
            NPRNT = 61
            IF(IFLG  .EQ. 0) GO TO 110
    66     WRITE(NPRNT,23)
    23     FORMAT(//                                    27X,*CODE    MEAN       STD.DEV.
          1*  MIN.        MAX.      OBS.*/)
            GO TO 120
    110     WRITE(NPRNT,1000)
   1000     FORMAT(/ 44X, *NON ZERO VALUES ONLY  *)
    120     XS = SUM1
            XSS = SUM2
            XN = SUM3
           AVG=XS/YN
           STD=(((XN*XSS)-(XS*XS))/(XN*(XN-1.0)))**.5
           N=XN
           WRITE(NPRNT,24)I,AVG,STD, SUM4,SUM5,N
    24     FORMAT(27X,I3,4F11.4,I7)
            IF(IFLG .EQ. 0 ) RETURN
            IFLG = 0
            GO TO 5
            END
C
C
          SUBROUTINE PAGE (IPAGE)
          COMMON/CTRL/ WEEK,DAY,SEM ,TIME,DT,ICASE,ETIME,STIME,NWEEK
          INTEGER       WEEK,DAY,SEM ,TIME,DT,ICASE,ETIME,STIME
          IDAY=IDATE(KL)
          WRITE(61,100)  IPAGE,IDAY, ICASE
  100     FORMAT(1H1, *PAGE NO.*,I5,7X,
         1                A8,20X,*MODEL TEACHER EDUCATION SIMULATION PROGRA
         220X, *CASE NUMBER *  I5)
           IPAGE = IPAGE + 1
          RETURN          $ END
C
C
          SUBROUTINE ERROR(I,A,B,C,D,E,F)
          COMMON/CTRL/ WEEK,DAY,SEM ,TIME,DT,ICASE,ETIME,STIME,NWEEK
          INTEGER      WEEK,DAY,SEM ,TIME,DT,ICASE,ETIME,STIME
          WRITE(61,1002), TIME,I,A,B,C,D,E,F
    1002  FORMAT (* ERROR  *I10,A6,6I10)
          WRITE(61,1001), TIME,I,A,B,C,D,E,F
    1001  FORMAT ( *  ERROR *,I10,A6,6F15.2)
          RETURN $ END
          SUBROUTINE TITLE
          IDAY=IDATE(KL)
          WRITE(61,4001) IDAY
    4001  FORMAT(1H1///////////////////////////
         1  18X, * MODEL ELEMENTARY TEACHER   *//
         2 23X, *EDUCATION PROGRAM * //
         3 18X, *COMPUTER SIMULATION PROGRAM * //
         4 20X,*DATE OF RUN  * A8           )
          RETURN $ END
C
C
            IDENT     IDATE
            ENTRY     IDATE
```

```
IDATE     UBJP       *,,*
          DATE
          SLJ        IDATE
          END
C
C
      SUBROUTINE GRAPH ( N,AM,A,B    ,AMAX, AMIN     ,DELTA     )
      COMMON/CTRL/ WEEK.DAY.SEM ,TIME,DT,ICASE,ETIME,STIME,NWEEK
      INTEGER      WEEK.DAY.SEM .TIME.DT,ICASE,ETIME,STIME
      DIMENSION       MARK(120),AM(N)    ,BIN(120)
      INTEGER , A, B
C                                  AM(I) = ARRAY TO BE GRAPHED
C                                   N = NO OF ELEMENTS IN A(N)
C                                   A  = PRINTER CHARACTER
C                                   B = PRINTER CHARACTER FILLER
C                                          MAX OF 120 LINES IN GRAPH
       KOUNT = 1
   28 PRINT 100
  100 FORMAT(1H0/////)
      PRINT 101
  101 FORMAT (13X,122(1H-))
      AK = AMAX
    4 DO 3 I =1,120
      MARK(I) = B
      IF (AM(I).GT.AK)   MARK(I)=A
    3 CONTINUE
      WRITE( 61, 1000)(AK, (MARK(I) , I =1,120)     )
 1000 FORMAT(* *F10.2,2X,1H1,120(A1),1H1)
       IF(AK .LE. .001) GO TO 500
      AK = AK-DELTA
       IF(AK .LT. 0.0) AK = .001
       IF(KOUNT .GT. 120) GO TO 500
      IF(AK .GE. AMIN  ) GO TO 4
  500 PRINT 102
  102 FORMAT (13X,*-*, 24(5H----*),2H--)
      PRINT 104,(K,K=5,120,5)
  104 FORMAT (6X,4HWEEK,4X,24I5,/)
      PRINT 105
  105 FORMAT (14X,8(14X,1H*) )
      PRINT 107,(J,J=1,8)
  107 FORMAT (6X,8HSEMESTER ,8I15)
       CALL SDIV(AM, TIME)
      RETURN
      END
C
C
```